AD-A039 784 LOCKHEED MISSILES AND SPACE CO INC PALO ALTO CALIF PA--ETC F/G 9/2
COMPUTATIONAL ASPECTS OF TIME INTEGRATION PROCEDURES IN STRUCTU--ETC(U)
JAN 77 K C PARK, C A FELIPPA N00014-74-C-0355

UNCLASSIFIED LMSC/D556247 NL

1 OF 2
AD
A039784

064-559
code 474

(11) JANUARY 1977 (12) 116p. (10) (12) (14) LMSC-D556247

(6)

COMPUTATIONAL ASPECTS OF TIME INTEGRATION PROCEDURES IN STRUCTURAL DYNAMICS.

PART I: IMPLEMENTATION.

by

C. A. Felippa
K. C. Park

PART II: ERROR PROPAGATION,

by

(10) K. C. Park
C. A. Felippa

Research Sponsored by the Independent Research Program of Lockheed Missiles & Space Co., Inc., and by the new Office of Naval Research under Contract (15) N00014-74-C-0355

Structures Laboratory
Org. 52-33, Bldg 205
Lockheed Palo Alto Research Laboratory
Palo Alto, California 94304

210 118

LMSC-D556247

COMPUTATIONAL ASPECTS OF TIME INTEGRATION PROCEDURES IN STRUCTURAL DYNAMICS

PART I: IMPLEMENTATION

C. A. Felippa
K. C. Park

January 1977

Structures Laboratory
Lockheed Palo Alto Research Laboratory
Palo Alto, California

ABSTRACT

A unified approach for the implementation of direct time integration procedures in structural dynamics is presented. Two key performance assessment factors are considered, viz., computational effort and error propagation. It is shown that these factors are strongly affected by details in the reduction of the second-order equations of motion to a system of first-order equations, and by the computational path followed at each time step. Part I is primarily devoted to the study of the organization of the computational process. Specific implementation forms derivable from the unified approach are studied in detail, and rated accordingly.

CONTENTS OF PART I

Section 1
INTRODUCTION

Motivation

The use of direct time integration procedures for the numerical treatment of structural dynamics problems has attracted an increasing number of investigators in recent years. Most studies have concentrated on the development of new methods, and on the formulation and verification of performance-evaluation criteria. These studies have established stability and accuracy (in that order) as the most important attributes of a time integration operator, and there is an emerging consensus on the use of evaluation criteria appropriate to the treatment of mechanical equations of motion (or, in fact, any stiff-oscillatory system). Accuracy can be assessed by measuring numerical damping and frequency distortion as a function of the sampling rate when the integrator is applied to undamped linear-oscillator models. Stability can be assessed by studying the amplitude growth of the computed solutions of linear and nonlinear model problems.

There is presently widespread agreement on the stability advantages of implicit operators for treating structural dynamics problems, in which the gross response of the discrete model is dominated by the low-frequency components of the modal spectrum. Explicit schemes are still widely used, however, in problems involving shock-type excitation and wave-propagation type response, where the contribution of the high-frequency components remain significant throughout the analysis process. Performance cross-over points between explicit and implicit schemes have not been firmly established as yet.

The algorithmic properties pertaining to stability and accuracy play an important role in the preliminary evaluation of time integration operators, and represent a useful source of information for selecting methods appropriate to different classes of problems. There have been conflicting reports, however, as regards the actual performance of the same integrator in different computer programs. Contradictory claims are especially frequent in nonlinear analysis. These unresolved questions suggest that the computer implementation of a time integration procedure has a substantial bearing on the performance of a transient analysis program. It will be shown that computer implementation details affect two important performance characteristics: the computational efficiency and the propagation of computational error.

Computational Efficiency

For a given stepsize sequence, the efficiency of a transient analyzer is largely controlled by the volume of arithmetic operations and data transfers performed at each step. These activities are governed in turn by the allocation of storage resources and by the order and manner in which array-processing operations are carried out. It will be shown that several ways of implementing the basic time-advancing cycle are possible even if the same integration operator is used, and that these implementations display varying efficiency characteristics.

Computational Error Propagation

Computational errors arise from three main sources:

- Rounding errors resulting from the use of finite-precision arithmetic. This source is significant if the linear or nonlinear algebraic equations that must be solved at each time station are ill-conditioned.
- Inaccuracies resulting from the approximate solution of the aforementioned equations by iterative techniques in the case of nonlinear problems.
- Errors in the calculation of applied forces. This source can be significant if the excitation depends on the solution of a coupled initial value problem, as in the case of fluid-structure interaction, dynamic thermal loads, and certain types of material nonlinearity.

The "local" computational error due to the combined effect of the preceding sources at a typical time station t_n may be viewed as a spurious excitation source. This disturbance is propagated, and usually amplified, by the integration operator to the stream of computed solutions for $t > t_n$. The effect of a local perturbation on downstream solutions will be called the propagated computational error. The overall computational error at a given time station results from the combination of the local error and of the propagated errors emanating from all previous stations; this quantity will be called the accumulated (or global) computational error.

A comprehensive theory of error propagation in the direct integration of nonstiff ordinary differential equations (ODE) has been laid out by Henrici [1, 2]. His investigations concentrated on the development of asymptotic bounds for the accumulated computational error, i. e., bounds applicable if $|\omega_i h| << 1$, where h is the integration stepsize and ω_i are characteristic roots of the associated homogeneous system. This paper complements and extends Henrici's results to the stiff ODE systems arising in structural dynamics applications, and studies the crucial effect of implementation details on error propagation.

Objectives of the Paper

The paper has two basic objectives. First, to present a general framework for reducing the second-order equations of structural dynamics to an equivalent first-order system, to which a pair of linear multistep (LMS) implicit time integrators is applied, and to categorize the resulting implementation variants. This approach is shown to produce many of the computer implementations in current favor, as well as a myriad of hitherto untried ones. Second, to study the dependence of the propagated computational error on various implementations resulting from the general approach, and to correlate that behavior with compositional properties of the integration operator.

Because of its length and varying degree of mathematical sophistication expected from the reader, the paper is subdivided into two parts, which address the preceding objectives in turn.

Outline of Part I

The reduction of the governing matrix equations of linear structural dynamics to a first-order system is presented following the auxiliary-vector procedure introduced by Jensen [3]. Two LMS implicit integration operators are applied to the resulting first-order system to derive a set of coupled difference equations. This set contains arbitrary weighting matrices associated with the choice of auxiliary vector. The selection of convenient weighting matrices leads to specific formulations, which in turn subdivide into variants according to the computational path followed in the typical time-advancing cycle. Operation counts associated with major array-manipulation tasks are exhibited as a function of the computer implementation, sparseness attributes of the mass and damping matrices, and certain compositional properties of the integration operators. Associated starting procedures that minimize the need for additional computations are described. Finally, the advantages and disadvantages of the various formulations are summarized.

Practical considerations on the *computational error propagation* behavior associated with various implementations are interspersed throughout Part I. These considerations represent an extract of relevant conclusions of the investigations reported in Part II, and are offered so as to make Part I relatively self-contained for those readers that are not particularly motivated to follow the details of the error analysis.

Linear dynamic systems are emphasized throughout, as this keeps the major points of the exposition uncluttered from unnecessary complexities brought about by the intrusion of nonlinear terms. Many of the conclusions regarding computer implementation aspects and error propagation behavior are applicable "locally" to the linearized forms of the governing nonlinear dynamic equations, whether resulting from the tangent-stiffness or the pseudo-force approach.

Section 2
SUMMARY OF NEW RESULTS IN PART I

1. Categorization of computer implementation procedures according to the selection of auxiliary vector for the reduction to a first-order system.

2. Distinction of three basic computational paths for the evaluation of auxiliary vectors in the time-advancing cycle.

3. Tabulation of computational effort associated with various implementations.

4. The concept of "starting operator family" to minimize the need for additional computations in the starting procedure.

5. Formal embedding of the special, two-derivative operators of structural mechanics (Newmark, Houbolt, Wilson) within the general implementation-classification approach.

Section 3
FORMULATION OF DIFFERENCE EQUATIONS

Governing Equations

The governing equations of motion of a linear, discrete mechanical system with n_f degrees of freedom may be written

$$\underset{\sim}{M}\,\ddot{\underline{u}} + \underset{\sim}{D}\,\dot{\underline{u}} + \underset{\sim}{K}\,\underline{u} = \underline{f}(t) \tag{1}$$

where $\underline{u} = \underline{u}(t)$ is a state vector containing n_f displacement coordinates defining the spatial configuration of the discrete system at each time t, $\underset{\sim}{M}$, $\underset{\sim}{D}$, and $\underset{\sim}{K}$ are the linear mass, damping, and stiffness matrices, respectively, and $\underline{f}(t)$ is a vector of time-varying applied generalized forces; in (1) a dot denotes temporal differentiation.

Reduction to a First-Order System

A general procedure for reducing (1) to a system of first-order equations has been presented by Jensen [3]. He introduced an auxiliary vector $\underline{v}$ defined as a weighted combination of displacements and velocities:

$$\underline{v} = \underset{\sim}{A}\,\underset{\sim}{M}\,\dot{\underline{u}} + \underset{\sim}{B}\,\underline{u} \tag{2}$$

where $\underset{\sim}{A}$ and $\underset{\sim}{B}$ are (for the moment) arbitrary n_f by n_f matrices with the proviso that $\underset{\sim}{A}$ be nonsingular. The resulting first-order system is

$$\begin{bmatrix} \underset{\sim}{A}\,\underset{\sim}{M} & \underset{\sim}{0} \\ \underset{\sim}{A}\,\underset{\sim}{D} - \underset{\sim}{B} & \underset{\sim}{I} \end{bmatrix} \begin{Bmatrix} \dot{\underline{u}} \\ \dot{\underline{v}} \end{Bmatrix} + \begin{bmatrix} \underset{\sim}{B} & -\underset{\sim}{I} \\ \underset{\sim}{A}\,\underset{\sim}{K} & \underset{\sim}{0} \end{bmatrix} \begin{Bmatrix} \underline{u} \\ \underline{v} \end{Bmatrix} = \begin{Bmatrix} \underline{0} \\ \underset{\sim}{A}\,\underline{f} \end{Bmatrix} \tag{3}$$

where $\underset{\sim}{I}$ and $\underset{\sim}{0}$ denote the identity and null matrix, respectively. Eq. (3) fits within the standard ODE-system format:

$$\dot{\underline{y}} = \underline{F}(\underline{y}, t) \tag{4}$$

where

$$\underline{y} = \left\{ \begin{matrix} \underline{u} \\ \underline{v} \end{matrix} \right\} \tag{5}$$

and

$$\underline{F} = \begin{bmatrix} \underset{\sim}{A}\,\underset{\sim}{M} & \underset{\sim}{0} \\ \underset{\sim}{A}\,\underset{\sim}{D} - \underset{\sim}{B} & \underset{\sim}{I} \end{bmatrix}^{-1} \left\{ \begin{matrix} \underline{v} - \underset{\sim}{B}\,\underline{u} \\ \underset{\sim}{A}\,(\underline{f} - \underset{\sim}{K}\,\underline{u}) \end{matrix} \right\} \tag{6}$$

Integration Method

We consider the numerical integration of the first-order system (3) using m-step, one-derivative, linear multistep (LMS) operators to discretize the time variation of $\underline{u}$ and $\underline{v}$. A detailed treatment of these methods can be found elsewhere, e. g., [1-2, 4-5]. For a constant stepsize h, LMS operators applied to Eq. (3) at the n-th time station t_n can be written

$$\left. \begin{aligned} \sum_{i=0}^{m} \alpha_i\, \underline{u}_{n-i} &= h \sum_{i=0}^{m} \beta_i\, \dot{\underline{u}}_{n-i} \\ \sum_{i=0}^{m} \gamma_i\, \underline{v}_{n-i} &= h \sum_{i=0}^{m} \delta_i\, \dot{\underline{v}}_{n-i} \end{aligned} \right\} \quad \begin{aligned} & m \geq 1,\ n \geq m \\ & \alpha_o \neq 0,\ \gamma_o \neq 0 \end{aligned} \tag{7}$$

where α_i, β_i, γ_i, and δ_i are scalars associated with specific finite difference operators, and $\underline{u}_k$, $\underline{v}_k$, ... stand for the vectors $\underline{u}(t_k)$, $\underline{v}(t_k)$, ... , computed through the discretization procedure (7). The above difference expressions may be scaled by arbitrary factors. In the sequel they will be normalized by requiring that

$$\alpha_o = \gamma_o = 1 \tag{8}$$

Remark. In the most general case one could use an m-step and a k-step (k < m) LMS operator in (7). The latter can always be considered as an m-step operator, however, by appropriately appending (m-k) zero coefficients.

With a view to subsequent manipulations, it is convenient to introduce a more compact notation than (7), which nevertheless separates explicitly the current state variables $\underline{u}_n$, $\underline{\dot{u}}_n$, $\underline{v}_n$, $\underline{\dot{v}}_n$ from historical terms involving past solutions:

$$\begin{aligned} \underline{u}_n + \underline{a}_n^u &= h_\beta \, \underline{\dot{u}}_n + h \, \underline{b}_n^{\dot{u}} \\ \underline{v}_n + \underline{c}_n^v &= h_\delta \, \underline{\dot{v}}_n + h \, \underline{d}_n^{\dot{v}} \end{aligned} \tag{9}$$

where the "generalized stepsizes" h_β, h_δ are defined by

$$h_\beta = h\,\beta_o \qquad h_\delta = h\,\delta_o \tag{10}$$

and vectors $\underline{a}$, $\underline{b}$, ... denote linear combinations of past solutions:

$$\left[\underline{a}_n^z \, \underline{b}_n^z \, \underline{c}_n^z \, \underline{d}_n^z \right] = \left[\underline{z}_{n-1} \; \cdots \; \underline{z}_{n-m} \right] \begin{bmatrix} \alpha_1 & \beta_1 & \gamma_1 & \delta_1 \\ \vdots & \vdots & \vdots & \vdots \\ \alpha_m & \beta_m & \gamma_m & \delta_m \end{bmatrix} \tag{11}$$

in which $\underline{z}$ is an arbitrary vector symbol, which may stand for $\underline{u}$, $\underline{\dot{u}}$, ... etc. The current state solution $(\underline{u}_n, \underline{v}_n)$ can be expressed from (9) in the form

$$\begin{Bmatrix} \underline{u}_n \\ \underline{v}_n \end{Bmatrix} = \begin{Bmatrix} h_\beta \, \underline{\dot{u}}_n \\ h_\delta \, \underline{\dot{v}}_n \end{Bmatrix} + \begin{Bmatrix} \underline{h}_n^u \\ \underline{h}_n^v \end{Bmatrix} \tag{12}$$

where

$$\begin{aligned} \underline{h}_n^u &= h \, \underline{b}_n^{\dot{u}} - \underline{a}_n^u \\ \underline{h}_n^v &= h \, \underline{d}_n^{\dot{v}} - \underline{c}_n^v \end{aligned} \tag{13}$$

will be called, following Jensen [3], the historical vectors pertaining to $\underline{u}$ and $\underline{v}$, respectively.

Implicit Discretization

We shall consider only implicit LMS operators, for which both β_o and δ_o are nonzero, whence $h_\beta \neq 0$ and $h_\delta \neq 0$. Elimination of $\dot{\underline{u}}_n$ and $\dot{\underline{v}}_n$ from (3) and (12) then produces the linear algebraic system of order $2n_f$:

$$\begin{bmatrix} \underset{\sim}{A}\,\underset{\sim}{M} + h_\beta \underset{\sim}{B} & -h_\beta \underset{\sim}{I} \\ \underset{\sim}{A}\,\underset{\sim}{D} - \underset{\sim}{B} + h_\beta \underset{\sim}{A}\,\underset{\sim}{K} & \eta I \end{bmatrix} \begin{Bmatrix} \underline{u}_n \\ \underline{v}_n \end{Bmatrix} = \begin{Bmatrix} \underset{\sim}{A}\,\underset{\sim}{M}\,\underline{h}^u_n \\ (\underset{\sim}{A}\,\underset{\sim}{D} - \underset{\sim}{B})\,\underline{h}^u_n + \eta \underline{h}^v_n + h_\beta \underset{\sim}{A}\,\underline{f}_n \end{Bmatrix} \qquad (14)$$

where $\eta = h_\beta / h_\delta = \beta_o / \delta_o$. Finally, elimination of $\underline{v}_n$ from (14) yields the n_f-th order linear system

$$\underset{\sim}{E}\,\underline{u}_n = \underline{g}_n \qquad (15)$$

where

$$\begin{aligned} \underset{\sim}{E} &= \underset{\sim}{M} + h_\delta \underset{\sim}{D} + h_\beta h_\delta \underset{\sim}{K} + (h_\beta - h_\delta)\,\underset{\sim}{C} \\ \underline{g}_n &= [\underset{\sim}{M} + h_\delta(\underset{\sim}{D} - \underset{\sim}{C})]\,\underline{h}^u_n + h_\beta \underset{\sim}{A}^{-1}\,\underline{h}^v_n + h_\beta h_\delta \underline{f}_n \qquad (16) \\ \underset{\sim}{C} &= \underset{\sim}{A}^{-1}\,\underset{\sim}{B} \end{aligned}$$

As $\underset{\sim}{M}$, $\underset{\sim}{D}$, and $\underset{\sim}{K}$ are normally symmetric, $\underset{\sim}{C} = \underset{\sim}{A}^{-1}\,\underset{\sim}{B}$ should also be a symmetric matrix if $h_\beta \neq h_\delta$ in order to avoid the appearance of an unsymmetric coefficient matrix $\underset{\sim}{E}$. This may pose certain constraints on the selection of $\underset{\sim}{A}$ and $\underset{\sim}{B}$.

The generality of the discretization (7), in which different LMS operators may be used for $\underline{u}$ and $\underline{v}$, has two important advantages. First, the use of different approximations for $\underline{u}$ and $\underline{v}$ is allowed, should that prove desirable on account of the physical meaning of $\underline{v}$ resulting from specific choices of $\underset{\sim}{A}$ and $\underset{\sim}{B}$ in (2). Second, special two-derivative integrators devised for treating the second-order system (1) directly, such as the widely used Newmark [6], Houbolt [7], and Wilson [8] operators, can be presented as special cases of a slight extension of (7), in which a historical $\overline{\underline{u}}$-term is appended to (7b), as shown in Appendix A. It follows that a separate study of the computer implementation of those special methods is not required.

The selection of a common LMS operator for both $\underline{u}$ and $\underline{v}$ is natural if the analyst wishes to keep similar discretization accuracy on both state vectors. For this particular choice, Eqs. (9) become

$$\begin{aligned} \underline{u}_n + \underline{a}_n^{u} &= h_\beta \dot{\underline{u}}_n + h\, \underline{b}_n^{\dot{u}} \\ \underline{v}_n + \underline{a}_n^{v} &= h_\beta \dot{\underline{v}}_n + h\, \underline{b}_n^{\dot{v}} \end{aligned} \tag{17}$$

and the components of (15) simplify slightly to

$$\begin{aligned} \underset{\sim}{E} &= \underset{\sim}{M} + h_\beta \underset{\sim}{D} + h_\beta^2 \underset{\sim}{K} \\ \underline{g}_n &= [\underset{\sim}{M} + h_\beta (\underset{\sim}{D} - \underset{\sim}{C})]\, \underline{h}_n^{u} + h_\beta \underset{\sim}{A}^{-1} \underline{h}_n^{v} + h_\beta^2 \underline{f}_n \end{aligned} \tag{18}$$

The coefficient matrix $\underset{\sim}{E}$ becomes independent of the choice of $\underset{\sim}{A}$ and $\underset{\sim}{B}$ in this case, as noted by Jensen [3].

Section 4
COMPUTER IMPLEMENTATION

Implementations of the implicit time integration procedure outlined in the preceding section may differ in two respects: (a) the selection of weighting matrices $\underset{\sim}{A}$ and $\underset{\sim}{B}$ in (2), and (b) the manner in which the resulting auxiliary vector $\underline{v}$ and its temporal derivative $\dot{\underline{v}}$ are computed at each step. We deal with the latter topic first before passing to discuss the selection of computationally convenient $\underset{\sim}{A}$ and $\underset{\sim}{B}$.

Computational Paths

Three basic computational paths, labelled (0), (1) and (2), may be followed in advancing the discrete solution over a typical time step. The associated sequences of calculations, for arbitrary $\underset{\sim}{A}$ and $\underset{\sim}{B}$, are flow charted in Figure 1.

The path-identification index (0-2) gives an idea of the number of backward difference operations performed in the determination of $\underline{v}_n$ and $\dot{\underline{v}}_n$ over each solution-advancing cycle. As shown in Part II, this index plays an important role in the study of the error propagation behavior of particular implementations.

Path (0) is consistent with the first-order difference system (14), i.e., the computed vectors satisfy (14) exactly if computational errors are ignored. The original ODE system (1) is also satisfied. On the other hand, the computed $\underline{u}_n$, $\dot{\underline{u}}_n$ and $\underline{v}_n$ do not, in general, verify the auxiliary-vector definition (2), which is satisfied only in the limit $h \rightarrow 0$.

A slight variant of path (0), labelled (0'), is also shown in Figure 1. In this variant the velocity vector $\dot{\underline{u}}_{n-1}$ is recomputed so that (2) is also verified at past stations. This additional operation is not only expensive for general $\underset{\sim}{A}$ and $\underset{\sim}{B}$, but worthless as it does not improve the error propagation characteristics of the integrator. It is included here, however,

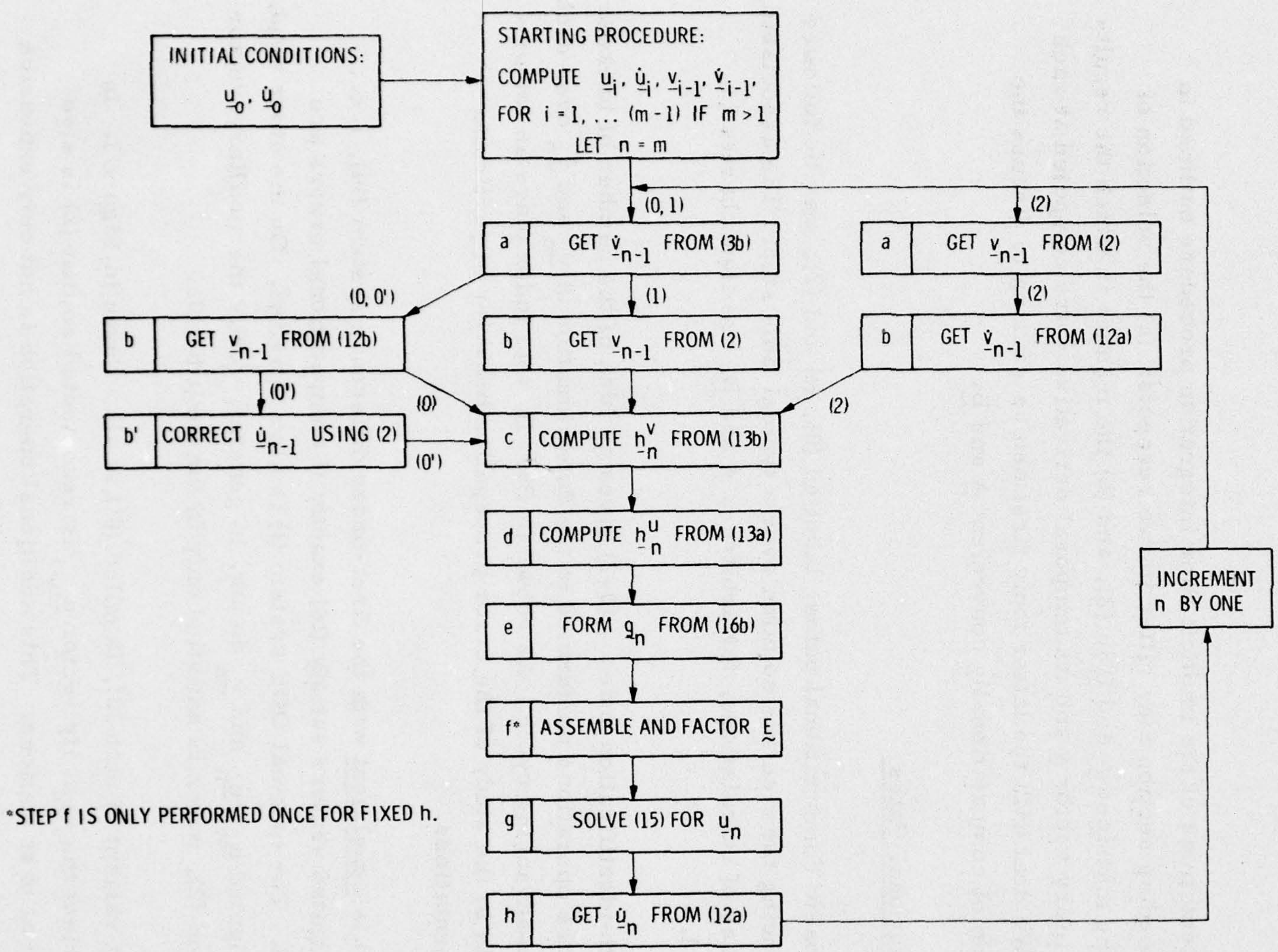

INITIAL CONDITIONS:
$\underset{\sim}{u}_o$, $\dot{\underset{\sim}{u}}_o$
STARTING PROCEDURE:
COMPUTE $\underset{\sim}{u}_i$, $\dot{\underset{\sim}{u}}_i$, $\underset{\sim}{v}_{i-1}$, $\dot{\underset{\sim}{v}}_{i-1}$,
FOR i = 1, ... (m - 1) IF m > 1
LET n = m
(0, 1)
a GET $\dot{\underset{\sim}{v}}_{n-1}$ FROM (3b)
(0, 0')
b GET $\underset{\sim}{v}_{n-1}$ FROM (12b)
(0')
b' CORRECT $\dot{\underset{\sim}{u}}_{n-1}$ USING (2)
(0)
(0')
(1)
b GET $\underset{\sim}{v}_{n-1}$ FROM (2)
c COMPUTE $\underset{\sim}{h}^v_n$ FROM (13b)
d COMPUTE $\underset{\sim}{h}^u_n$ FROM (13a)
e FORM $\underset{\sim}{g}_n$ FROM (16b)
f* ASSEMBLE AND FACTOR $\underset{\sim}{E}$
g SOLVE (15) FOR $\underset{\sim}{u}_n$
h GET $\dot{\underset{\sim}{u}}_n$ FROM (12a)
(2)
a GET $\underset{\sim}{v}_{n-1}$ FROM (2)
(2)
b GET $\dot{\underset{\sim}{v}}_{n-1}$ FROM (12a)
(2)
INCREMENT n BY ONE
*STEP f IS ONLY PERFORMED ONCE FOR FIXED h.

Fig. 1 Computational paths for arbitrary auxiliary vector

because it occurs naturally in the conventional choice $\underline{v} = \underline{\dot{u}}$ discussed below.

Path (1) satisfies both differential expressions (1) and (2) at the expense of (14). Finally, path (2) enforces the $\underline{v}$-definition (2) and the difference system (14) at the expense of the ODE system (1).

Remark. It should not be surmised that the flow chart of Figure 1 exhausts all possible computational paths. There is in fact another path family in which steps (c, d) are carried out prior to (a, b). The practical implementation of these sequence types is tied up, however, to very specific choices of $\underset{\sim}{A}$ and $\underset{\sim}{B}$, which become (linear) functions of the stepsize h. A study of these alternative implementations, which offer certain organizational advantages, will appear in a sequel article.

Selection of Auxiliary Vector

The behavior of the propagated computational error is independent of the selection of auxiliary vector $\underline{v}$ through Eq. (2), but the computational effort per step is not. For certain choices of $\underset{\sim}{A}$ and $\underset{\sim}{B}$ the volume of computations--and, to a lesser extent, the storage requirements--can be significantly reduced because some of the computational steps displayed in Figure 1 can be either simplified or bypassed entirely. The actual operation counts turn out to depend on sparseness characteristics of the matrices $\underset{\sim}{M}$ and $\underset{\sim}{D}$, as well as on certain properties of the integration operators, and are discussed in a later section. We proceed next to examine some computationally convenient choices of $\underset{\sim}{A}$ and $\underset{\sim}{B}$, which will be henceforth identified as formulations or "forms" for short.

Conventional Formulation (C-Form)

The choice of $\underline{v}$ described in standard textbooks--e.g., [1, 2, 4, 5]--and in most papers dealing with second-order systems is

$$\underline{v} = \underline{\dot{u}} \tag{19}$$

which corresponds to

$$\underset{\sim}{A} = \underset{\sim}{M}^{-1}, \quad \underset{\sim}{B} = \underset{\sim}{C} = \underset{\sim}{0} \tag{20}$$

The temporal derivative $\dot{\underline{v}}$ is simply the acceleration vector $\ddot{\underline{u}}$. With this choice, the components of the algebraic system (15) become

$$\begin{aligned} \underset{\sim}{E} &= \underset{\sim}{M} + h_\delta \underset{\sim}{D} + h_\delta h_\beta \underset{\sim}{K} \\ \underline{g}_n &= (\underset{\sim}{M} + h_\delta \underset{\sim}{D})\, \underline{h}^{u}_n + h_\beta \underset{\sim}{M}\, \underline{h}^{\dot{u}}_n + h_\delta h_\beta \underline{f}_n \end{aligned} \tag{21}$$

The sequence of calculations corresponding to paths (0'), (1) and (2) of Figure 1 are listed in Table 1, where they are identified as (C0'), (C1) and (C2), respectively.

The fact that the choice (19) effectively compresses the four state vectors $(\underline{u}_n, \dot{\underline{u}}_n, \underline{v}_n, \dot{\underline{v}}_n)$ into three $(\underline{u}_n, \dot{\underline{u}}_n, \ddot{\underline{u}}_n)$ leads to some simplifications in the general computational sequences of Figure 1. Some of the steps become trivial, but have been left in Table 1 for compatibility with the arrangement of Figure 1. Some practical considerations pertaining to the computational sequences of Table 1 follow.

1. <u>(C0') Form.</u> The determination of accelerations in step (a) requires that the mass matrix $\underset{\sim}{M}$ be nonsingular. This is an unfortunate restriction, as many commonly used mass-lumping procedures in finite element programs assign zero mass to non-translational degrees of freedom (the assignation of small "fictitious" masses does not solve the problem because $\underset{\sim}{M}$ remains ill-conditioned). If $\underset{\sim}{M}$ is nondiagonal, step (a) is computationally expensive.

2. <u>(C1) Form.</u> The appearance of $\underset{\sim}{M}^{-1}$ is avoided by working with the inertial force term $\underset{\sim}{M}\,\ddot{\underline{u}}$ throughout; consequently, $\underset{\sim}{M}$ can be singular.

3. <u>(C2) Form.</u> The requirement of nonsingular $\underset{\sim}{M}$ applies only to the starting procedure, during which the initial acceleration

$$\dot{\underline{v}}_o = \ddot{\underline{u}}_o = \underset{\sim}{M}^{-1} (\underline{f}_o - \underset{\sim}{D}\,\dot{\underline{u}}_o - \underset{\sim}{K}\,\underline{u}_o) \tag{22}$$

Table 1

COMPUTATIONAL SEQUENCES ASSOCIATED WITH THE CONVENTIONAL (C) FORMULATION $\underline{v} = \dot{\underline{u}}$

(C0')	a	$\ddot{\underline{u}}_{n-1} = M^{-1} (\underline{f}_{n-1} - D\,\dot{\underline{u}}_{n-1} - K\,\underline{u}_{n-1})$
	b, b'	$\dot{\underline{u}}_{n-1} = \underline{h}^{\dot{u}}_{n-1} + h_{\delta}\,\ddot{\underline{u}}_{n-1}$
	c	$\underline{h}^{\dot{u}}_{n} = h\,\underline{d}^{\ddot{u}}_{n} - \underline{c}^{\dot{u}}_{n}$
	d	$\underline{h}^{u}_{n} = h\,\underline{b}^{\dot{u}}_{n} - \underline{a}^{u}_{n}$
	e	$\underline{g}_{n} = (\underset{\sim}{M} + h_{\delta}\,\underset{\sim}{D})\,\underline{h}^{u}_{n} + h_{\beta}\,\underset{\sim}{M}\,\underline{h}^{\dot{u}}_{n} + h_{\beta}\,h_{\delta}\,\underline{f}_{n}$
	f	$\underset{\sim}{E} = \underset{\sim}{M} + h_{\delta}\,\underset{\sim}{D} + h_{\beta}\,h_{\delta}\,\underset{\sim}{K}$
	g	$\underline{u}_{n} = \underset{\sim}{E}^{-1}\,\underline{g}_{n}$
	h	$\dot{\underline{u}}_{n} = (\underline{u}_{n} - \underline{h}^{u}_{n})/h_{\beta}$
(C1)	a	$\underset{\sim}{M}\,\ddot{\underline{u}}_{n-1} = \underline{f}_{n-1} - \underset{\sim}{D}\,\dot{\underline{u}}_{n-1} - \underset{\sim}{K}\,\underline{u}_{n-1}$
	b	$\dot{\underline{u}}_{n-1} = \dot{\underline{u}}_{n-1}$ (trivial)
	c	$\underset{\sim}{M}\,\underline{h}^{\dot{u}} = h\,\underline{d}^{M\ddot{u}}_{n} - \underset{\sim}{M}\,\underline{c}^{\dot{u}}_{n}$
	d-h	Same as (C0')
(C2)	a	$\dot{\underline{u}}_{n-1} = \dot{\underline{u}}_{n-1}$ (trivial)
	b	Skipped
	c-g	Same as (C0')
	$\hat{h}$	$\ddot{\underline{u}}_{n} = (\underline{u}_{n} - \underline{h}^{u}_{n} - h_{\beta}\,\underline{h}^{\dot{u}}_{n})/(h_{\beta}\,h_{\delta})$
	h	$\dot{\underline{u}}_{n} = \underline{h}^{\dot{u}}_{n} + h_{\beta}\,\ddot{\underline{u}}_{n}$

must be calculated from the initial conditions $\underline{u}_o$, $\dot{\underline{u}}_o$. The calculation sequence shown in Table 1 differs slightly from that of Figure 1 in that the accelerations $\ddot{\underline{u}}_n$ are determined (through a double-differencing operation) prior to the calculation of velocities. The sequence in Table 1 is used more extensively in practice (especially in undamped systems) than that of Figure 1 but the numerical behavior of both implementations is identical.

Note that formulation (C0) is not shown. The implementation of path (0) always demands that $\underline{v}$ and $\dot{\underline{u}}$ be kept as separate arrays and is consequently incompatible with the compression to three state vectors ($\underline{u}_n$, $\dot{\underline{u}}_n$, $\ddot{\underline{u}}_n$) induced by the choice of (19).

Jensen's Formulation (J-Form)

Jensen [3] has advocated the selection

$$\underset{\sim}{A} = \underset{\sim}{I} \ , \quad \underset{\sim}{B} = \underset{\sim}{C} = \underset{\sim}{D} \tag{23}$$

which gives

$$\begin{aligned} \underline{v} &= \underset{\sim}{M}\dot{\underline{u}} + \underset{\sim}{D}\underline{u} \\ \dot{\underline{v}} &= \underset{\sim}{M}\ddot{\underline{u}} + \underset{\sim}{D}\dot{\underline{u}} = \underline{f} - \underset{\sim}{K}\underline{u} \end{aligned} \tag{24}$$

The auxiliary vector $\underline{v}$ can be interpreted as an array of generalized momenta. Its temporal derivative $\dot{\underline{v}}$ represents the unbalanced dynamic force, sometimes called the "impulse" in classical mechanics.

With the choice (23), Eqs. (16) become

$$\begin{aligned} \underset{\sim}{E} &= \underset{\sim}{M} + h_\beta \underset{\sim}{D} + h_\delta h_\beta \underset{\sim}{K} \\ \underline{g}_n &= \underset{\sim}{M}\underline{h}^u_n + h_\beta \underline{h}^v_n + h_\delta h_\beta \underline{f}_n \end{aligned} \tag{25}$$

If the damping matrix $\underset{\sim}{D}$ does not vanish, (25b) is slightly simpler than (21b).

Table 2

COMPUTATIONAL SEQUENCES ASSOCIATED WITH THE JENSEN (J) FORMULATION $\underline{v} = \underset{\sim}{M}\,\dot{\underline{u}} + \underset{\sim}{D}\,\underline{u}$

(J0)	a	$\dot{\underline{v}}_{n-1} = \underline{f}_{n-1} - \underset{\sim}{K}\,\underline{u}_{n-1}$
	b	$\underline{v}_{n-1} = \underline{h}^{v}_{n-1} + h_{\delta}\,\dot{\underline{v}}_{n-1}$
	c	$\underline{h}^{v}_{n} = h\,\underline{d}^{\dot{v}}_{n} - \underline{c}^{v}_{n}$
	d	$\underline{h}^{u}_{n} = h\,\underline{b}^{\dot{u}}_{n} - \underline{a}^{u}_{n}$
	e	$\underline{g}_{n} = \underset{\sim}{M}\,\underline{h}^{u}_{n} + h_{\beta}\,\underline{h}^{v}_{n} + h_{\delta}\,h_{\beta}\,\underline{f}_{n}$
	f	$\underset{\sim}{E} = \underset{\sim}{M} + h_{\beta}\,\underset{\sim}{D} + h_{\delta}\,h_{\beta}\,\underset{\sim}{K}$
	g	$\underline{u}_{n} = \underset{\sim}{E}^{-1}\,\underline{g}_{n}$
	h	$\dot{\underline{u}}_{n} = (\underline{u}_{n} - \underline{h}^{u}_{n})/h_{\beta}$
(J2)	a	$\underline{v}_{n-1} = \underset{\sim}{M}\,\dot{\underline{u}}_{n-1} + \underset{\sim}{D}\,\underline{u}_{n-1}$
	b	$\dot{\underline{v}}_{n-1} = (\underline{v}_{n-1} - \underline{h}^{v}_{n-1})/h_{\delta}$
	c-h	Same as (J0)

The sequence of computations corresponding to paths (0) and (2) of Figure 1 are displayed in Table 2, where they are identified as the (J0) and (J2) formulations, respectively. Formulations (J0') and (J1) are not shown as they lack any practical value (the verification of this assertion is left as an exercise to the interested reader).

In both J-forms, the presence of a singular $\underset{\sim}{M}$ does not cause any particular difficulty. There are also no special problems associated with the starting procedure.

Section 5

COMPARISON OF IMPLEMENTATION PROCEDURES

Computational Effort

In order to make an *a priori* comparison of the computational effort involved in specific implementations, it is necessary to make some simplifying assumptions:

1. The computational effort per step required to integrate a linear dynamic system with fixed h is considered to be dominated by *matrix/vector* operations. Operations involving only vectors are neglected. Diagonal matrices are considered equivalent to vectors. The assembly and factorization of $\underset{\sim}{E}$--step (f) in Figure 1--is not included in the count, as this operation is performed only once for fixed h and its effect on all formulations is identical. Similarly, the effort spent in starting and stepsize-change procedures is ignored.

2. The unit of work (U) is a *stiffness-matrix/vector multiply*, e. g., $\underset{\sim}{K}\,\underset{\sim}{u}_n$. Matrices $\underset{\sim}{M}$ and $\underset{\sim}{D}$ are assumed to be either diagonal ($\underset{\sim}{D}$ may be null) or have the sparseness structure of $\underset{\sim}{K}$. It follows that the $\underset{\sim}{E}$ matrix (21a) or (25a) has the sparseness structure of $\underset{\sim}{K}$. The solution of the linear system (15) is counted as 2U (one unit for the forward pass and one for the backsolve). This value is accurate only for fairly dense systems, but is not far off for matrices stored in a "skyline" or variable-bandwidth format.

3. In the case of large discrete systems that require out-of-core processing, the number of work units gives an idea of the number of passes of block-partitioned matrices through core buffers. On many computer systems, these I/O activities may dominate the total run cost.

4. The effort involved in the evaluation of the applied force vector $\underline{f}(t)$ is not considered, as this is not implementation dependent.

Table 3 summarizes the work per step associated with the implementations described in the preceding section. Work units are tabulated as a function of commonly occurring sparseness attributes of $\underset{\sim}{M}$ and $\underset{\sim}{D}$ as well as of certain properties of the integrator discussed in the following subsection. Optimal formulations, in the sense noted in the table, are given for each case. It is clear that the (J0) formulation is preferable in the majority of situations.

Backward Difference Operators

A backward difference (BD) operator is one possessing no "historical" temporal derivative terms. For instance, $\beta_i = 0$ $(i = 1, \ldots m)$ in Eq. (7a). BD operators represent a subset of the general class of LMS operators that is widely used to treat stiff differential systems. Important members of that subset include Gear's stiffly-stable family [4] (which is A-stable for m = 1, 2), and the 3-step, A-stable Park's method [9].

As indicated in Table 3, considerable reductions in the computational effort per step can be realized in most formulations if a pair of BD operators is used in Eq. (7). For example, in the (C1) and (C2) forms, the calculation of acceleration terms is no longer required. The beneficial effect on vector storage requirements is discussed in Appendix B.

Error Propagation Behavior

The following considerations represent a synopsis of part of the material expounded in Part II, which is selected insofar as it impacts the rating of the formulations listed in Table 3.

It is well known that the response $\underline{u}(t)$ of the linear differential system (1) can be expressed as the superposition of normal components pertaining to the frequencies ω_i and mode shapes $\underline{x}_i$ of the associated vibration eigenproblem

Table 3

COMPARISON OF COMPUTATIONAL EFFORT REQUIRED BY VARIOUS FORMULATIONS

Form	Amplification of Modal Error Source For $\Omega = \omega h << 1$	Singular $\underset{\sim}{M}$ Allowed?	Work Units Per Fixed -h Advancing Step							
			General LMS Operators				BD Operators ($\underline{b} \equiv \underline{d} \equiv \underline{0}$)			
			Zero/Diag. $\underset{\sim}{D}$		General $\underset{\sim}{D}$		Zero/Diag. $\underset{\sim}{D}$		General $\underset{\sim}{D}$	
			Diag. $\underset{\sim}{M}$	Gen. $\underset{\sim}{M}$	Diag. $\underset{\sim}{M}$	Gen. $\underset{\sim}{M}$	Diag. $\underset{\sim}{M}$	Gen. $\underset{\sim}{M}$	Diag. $\underset{\sim}{M}$	Gen. $\underset{\sim}{M}$
(C0')	O(1)	No	3	6	5	8	3	6	5	8
(C1)	$O(\Omega^{-1})$	Yes	3	4	5	6	2	3	3	4
(C2)	$O(\Omega^{-1})$	No***	2	3	3	4	2	3	3	4
(J0)	O(1)	Yes	3	4	3	4	3	4	3	4
(J2)	$O(\Omega^{-1})$	Yes	2	4	3	5	2	4	3	5
Optimal Forms	Error Propagation Behavior Important		(J0)	(J0)	(J0)	(J0)	(J0)	(J0)	(J0)	(J0)
	Computational Effort Important*		(J2)	(C2,** J0)	(J0)	(J0)	(C1)	(C1)	(J0)	(J0)

NOTES:

* For minimal work ties, the formulation with better error propagation control is preferred. In case of a two-way tie, the formulation allowing a singular $\underset{\sim}{M}$ is preferred.

** Only if $\underset{\sim}{M}$ is nonsingular (and well-conditioned), else take second choice.

*** Nonsingular $\underset{\sim}{M}$ required in starting procedure. This constraint could be relaxed by starting with (C1) and switching later to (C2), at the cost of added program complexity.

$$\left(\omega_i^2 \underset{\sim}{M} + \omega_i \underset{\sim}{D} + \underset{\sim}{K}\right) \underline{x}_i = 0 \qquad (26)$$

A similar decomposition applies to the local computational error sources, and to the propagation and accumulation of such sources.

Discrete element dynamic models of mechanical systems generally fall into the category of stiff-oscillatory* differential systems. This term is used to mean that the eigenfrequencies ω_i lie close to (or on) the imaginary axis, and that the ratio $|\omega_{max}|/|\omega_{min}|$ spans several orders of magnitude; typically 10^4 to 10^8. If these systems are treated by implicit time integration procedures, an A-stable scheme [4] is normally required.

For most problems in linear structural dynamics, the energy content characteristics of the excitation $\underline{f}(t)$--i.e., its energy frequency spectrum--is such that the response $\underline{u}(t)$ can be adequately represented by the superposition of a set of normal responses pertaining to low frequency modes. Accordingly, the stepsize h is chosen† so that the circular sampling frequencies $\Omega_i = \omega_i h$ vary from $\Omega << 1$ (corresponding to normal components to be accurately traced) to $\Omega >> 1$ (corresponding to normal components to be "filtered out" by the integrator). (A more precise stepsize classification scheme is discussed in Part II.)

The propagation of a modal error source to downstream solutions depends on three factors: the sampling frequency Ω, the computational path followed, and certain properties of the integration operator. It turns out that the amplification of a local error can only be significant in the low-frequency ($\Omega << 1$) and high-frequency ($\Omega >> 1$) regions of the modal spectrum.

In the low-frequency end ($\Omega << 1$), the error propagation is primarily affected by the computational path. Pertinent results are summarized in Table 3. If path (0)--or its variant (0')--is followed, the downstream

*The qualifier "stiff" is used here to denote a mathematical property (widely different time constants) and is unrelated to mechanical stiffness.

†In practice, the stepsize h is chosen adaptively by the integration program, which "senses" the characteristics of the excitation through accuracy-monitoring procedures.

amplification of a modal error source is asymptotically independent of the stepsize. On the other hand, if paths (1) or (2) are followed, local modal errors are amplified by a factor asymptotically proportional to Ω^{-1}. For example, if $\Omega = \omega h = 0.01$ (a common sampling rate for the fundamental frequency), the error amplification difference between paths (0) and (1, 2) is two orders of magnitude, and this result holds regardless of the particular integrators selected in (7).

The error amplification behavior in the high-frequency region $\Omega >> 1$ is controlled by the computational path followed and by the coefficients β_i and δ_i of the integration operators (7). As high-frequency error propagation is relatively unimportant in linear structural dynamics, it has not been included as a rating factor in Table 3.

Section 6
STARTING PROCEDURES

The actual implementation of the starting procedure for LMS operators with $m \geq 2$ is rarely mentioned in the literature. The most common procedure consists of using a one-step method, such as the trapezoidal rule, to generate (m-1) initial solutions; after that the program switches to the standard integrator. This approach usually requires one extra factorization, because the coefficient matrix (16a) of the starter does not necessarily coincide with that pertaining to the integrator (7). We describe here a technique that avoids the additional factorization.

Introduce the "starting family" of LMS operators:

$$\begin{aligned}
&\underline{u}_1 + \underline{a}_1^{u} = h_\beta \dot{\underline{u}}_1 + h\,\underline{b}_1^{\dot{u}} && \text{(1-step)}\\
&\underline{u}_2 + \underline{a}_2^{u} = h_\beta \dot{\underline{u}}_2 + h\,\underline{b}_2^{\dot{u}} && \text{(2-step)}\\
&\quad\cdots\cdots\\
&\underline{u}_m + \underline{a}_m^{u} = h_\beta \dot{\underline{u}}_m + h\,\underline{b}_m^{\dot{u}} && \text{(target operator)}
\end{aligned} \tag{27}$$

which are characterized by identical values of the generalized timestep h_β. A similar family is constructed for $\underline{v}$ if $h_\beta \neq h_\delta$. As an example, if the target integrator for $\underline{u}$ is the 3-step Park's method [9], for which $h_\beta = 0.6\ h$, a convenient starting family is

$$\begin{aligned}
&\underline{u}_1 - \underline{u}_o = 0.6\,h\,\dot{\underline{u}}_1 + 0.4\,h\,\dot{\underline{u}}_o\\
&\underline{u}_2 - 1.2\,\underline{u}_1 + 0.2\,\underline{u}_0 = 0.6\,h\,\dot{\underline{u}}_2 + 0.2\,h\,\dot{\underline{u}}_1\\
&\underline{u}_n - 1.5\,\underline{u}_{n-1} + 0.6\,\underline{u}_{n-2} - 0.1\,\underline{u}_{n-3} = 0.6\,h\,\dot{\underline{u}}_n\,,\ n = 3, \ldots
\end{aligned} \tag{28}$$

The two auxiliary formulas (28a, b) are A-stable [(28b) is obtained by combining 40% of the trapezoidal rule with 60% of Gear's two-step integrator]; hence, no spurious high-frequency oscillations can be triggered by the starter.

A change in the stepsize h requires an additional assembly and factorization of $\underset{\sim}{E}$ in most existing implementations. A matrix-scaling technique can be used, however, to eliminate such refactorizations in linear or nonlinear problems [10].

Section 7
CONCLUSIONS

A general framework for categorizing computer implementations of direct time integration procedures based on the treatment of an equivalent first-order system by implicit LMS operators has been presented. Five particular implementations associated with computationally convenient choices of auxiliary vector and with certain calculation sequences have been studied in some detail. It is appropriate to summarize in this final section the relative merits and disadvantages of those specific formulations.

The (C0') form has excellent error propagation characteristics for low-frequency components, but is substantially more expensive than the other forms if either $\underset{\sim}{M}$ or $\underset{\sim}{D}$ is nondiagonal. The explicit determination of accelerations requires the presence of a well-conditioned mass matrix. As this form is in no case superior to (J0), it is not recommended for practical use.

Both (Cl) and (C2) display poor error propagation behavior as regards low-frequency response components, since local errors are magnified by an amount inversely proportional to the stepsize h. (It should be noted that for many of the integration operators studied in Part II, this amount is larger if the (C2) form is used.) The (Cl) form has the advantage of not being affected by a singular (or ill-conditioned) mass matrix, because the product $\underset{\sim}{M}\,\underline{\ddot{u}}$ can be kept throughout the advancing cycle. Implementations based on (C2) may run into starting problems if $\underset{\sim}{M}$ is singular or ill-conditioned, as is often the case in finite element models that include rotational degrees of freedom. If a BD operator pair is used in (7), these two forms are computationally equivalent, and the preceding remarks do not apply.

The (J0) form emerges as an easy winner for a general purpose implementation. It has excellent error control, is not affected by the condition of the mass matrix, and minimizes the computational effort per step in the general damped case.

The (J2) form can be considered a convenient adjunct of (J0). It is easy to implement both (J) forms in the same computer program, since their storage arrangements (cf. Appendix B) are similar, and a simple branch takes care of the difference in steps (a,b) of Table 2. The (J2) form is computationally optimal in the frequent case of an undamped, lumped-mass model, and could be used in such instance if the analyst is not particularly concerned about error propagation characteristics.

Appendix A
TWO-DERIVATIVE METHODS

In the discussion of numerical integration procedures for second- and higher-order ODE systems, it has been customary--see, e.g., [1,4]--to categorize LMS methods as follows:

(I) The original equations are recast into a first-order system, which is treated by standard (one-derivative) LMS operators. This is the approach followed in the main body of the paper.

(II) LMS formulas containing higher-order derivatives are applied directly to the higher-order system.

For the second-order dynamic equations (1), approach (II) requires the introduction of an LMS operator containing the accelerations $\ddot{\underline{u}}$. A general form of such schemes has been studied by Geradin [11]. His expression, rewritten slightly to conform to the notation style of Eq. (7), is

$$\left.\begin{aligned} \sum_{i=0}^{m} \alpha_i \underline{u}_{n-i} &= h \sum_{i=0}^{m} \beta_i \dot{\underline{u}}_{n-i} \\ \sum_{i=0}^{m} \gamma_i \underline{u}_{n-i} &= \eta h^2 \sum_{i=0}^{m} \beta_i \ddot{\underline{u}}_{n-i} \end{aligned}\right\} \begin{aligned} & m > 1,\ n \geq m \\ & \alpha_o = \gamma_o = 1 \end{aligned} \qquad \text{(A1)}$$

where η is a scalar used to effect the normalization $\gamma_o = 1$ in (A1b), and $\beta_o \neq 0$ for implicitness. The identification of the β_i coefficients in both right-hand sides follows from elementary consistency considerations.

Explicit expressions of the Houbolt, Newmark, and Wilson operators in the multistep form (A1) may be found in [11]; some specific examples are given in Appendix C (after Part II). Eqs. (A1) may be rewritten using a compact notation similar to Eqs. (9):

$$\begin{aligned} \underline{u}_n + \underline{a}_n^u &= h_\beta \dot{\underline{u}}_n + h\, \underline{b}_n^{\dot{u}} \\ \underline{u}_n + \underline{c}_n^u &= \eta h\, h_\beta \ddot{\underline{u}}_n + \eta h^2 \underline{b}_n^{\ddot{u}} \end{aligned} \qquad \text{(A2)}$$

Substracting (A2a) from (A2b) to eliminate $\underline{u}_n$ and collecting historical terms, the LMS pair (A2) may be presented in a form analogous to (12):

$$\begin{Bmatrix} \underline{u}_n \\ \underline{v}_n \end{Bmatrix} = \begin{Bmatrix} h_\beta \dot{\underline{u}}_n \\ h_\delta \dot{\underline{v}}_n \end{Bmatrix} + \begin{Bmatrix} \underline{h}_n^u \\ \tilde{\underline{h}}_n^v \end{Bmatrix} \qquad \text{(A3)}$$

where

$$\begin{aligned} &\underline{v}_n = \dot{\underline{u}}_n \;, \quad \dot{\underline{v}}_n = \ddot{\underline{u}}_n \;, \quad h_\delta = \eta h \\ &\beta_o \tilde{\underline{h}}_n^v = h_\delta \underline{b}_n^{\ddot{u}} - \underline{b}_n^u + \left(\underline{a}_n^u - \underline{c}_n^u\right)/h \end{aligned} \qquad \text{(A4)}$$

Substitution of (A3) into (16) gives

$$\begin{aligned} \underset{\sim}{E} &= \underset{\sim}{M} + h_\delta \underset{\sim}{D} + h_\beta h_\delta \underset{\sim}{K} \\ \underline{g}_n &= \underset{\sim}{M}\left(h_\delta h\, \underline{b}_n^{\ddot{u}} - \underline{c}_n^u\right) + h_\delta \underset{\sim}{D}\, \underline{h}_n^u + h_\beta h_\delta \underline{f}_n \end{aligned} \qquad \text{(A5)}$$

If $\underset{\sim}{D} = \underset{\sim}{0}$, the algebraic system (15) reduces to

$$(\underset{\sim}{M} + h_\beta h_\delta \underset{\sim}{K})\, \underline{u}_n = \underset{\sim}{M}\left(h_\delta h\, \underline{b}_n^{\ddot{u}} - \underline{c}_n^u\right) + h_\beta h_\delta \underline{f}_n \qquad \text{(A6)}$$

which expresses the well-known fact that the approximations of velocities through (A1a) is not required for the direct integration of an <u>undamped</u> system.

The implementation of these methods is seen to be inextricably linked to the conventional choice (19) of auxiliary vector. Accordingly, the three computational paths (0', 1, 2) of Table 1 may be used (with appropriate modifications to account for the expression (A5b) of $\underline{g}_n$). But it is not difficult to show that (C0') and (C1) are computationally identical in the case of an undamped system, and very similar if there is slight damping. Hence, only two implementation variants need to be considered: (C0'/C1) and (C2).

Both implementations turn out to display local error amplification of order h^{-1} for low-frequency components, i.e., the beneficial effect of the (C0') form on error propagation control is lost if a two-derivative operator (A1b) is used. As (C0'/C1) consistently involves equal or larger computational effort per step than (C2)--cf. Table 3--the latter emerges as a clear choice.

Henrici has described [1, Ch. 6] a radical modification of (A1b), called the "summed form," which eliminates the $O(h^{-1})$ dependence of the propagated local error. For an undamped ($\underset{\sim}{D} = \underset{\sim}{0}$) system (1), this device is equivalent to using the reduced first-order form:

$$\begin{Bmatrix} \underset{\sim}{M}\,\dot{\underline{u}} \\ \underset{\sim}{M}\,\dot{\underline{v}} \end{Bmatrix} = \begin{Bmatrix} \underset{\sim}{M}\,\underline{v} \\ \underline{f} - \underset{\sim}{K}\,\underline{u} \end{Bmatrix} \tag{A7}$$

Eq. (A7a) is treated by a standard LMS operator derived from (A1b) by formal division of the polynomial $\gamma_o z^m + \ldots + \gamma_m$ by $(z-1)$ while (A7b) is treated by a backward-Euler operator. This technique essentially amounts to a splitting of (A1b) into the formal product of two one-derivative operators. Therefore, effective control of computational error propagation demands a reduction to a first-order system, whether explicitly done as in (3) or concealed under a special name.

Appendix B
ARRAY STORAGE REQUIREMENTS

This Appendix should be of interest to implementors involved in the programming of one of the formulations defined in Tables 1 and 2. Table 4 provides a "snapshot" view of array structures involved in forms (Cl) and (J0), which are chosen to typify the (C) and (J) formulations, respectively. The flow of computations required to advance the solution over the current (n-th) step is indicated with arrowed paths. Matrix storage is not explicitly shown, as this is formulation-independent.

If the number of degrees of freedom (n_f) does not exceed a few thousands, all vectors displayed in Table 4 can be kept in high-speed memory (core) throughout. Vector storage counts in such case are

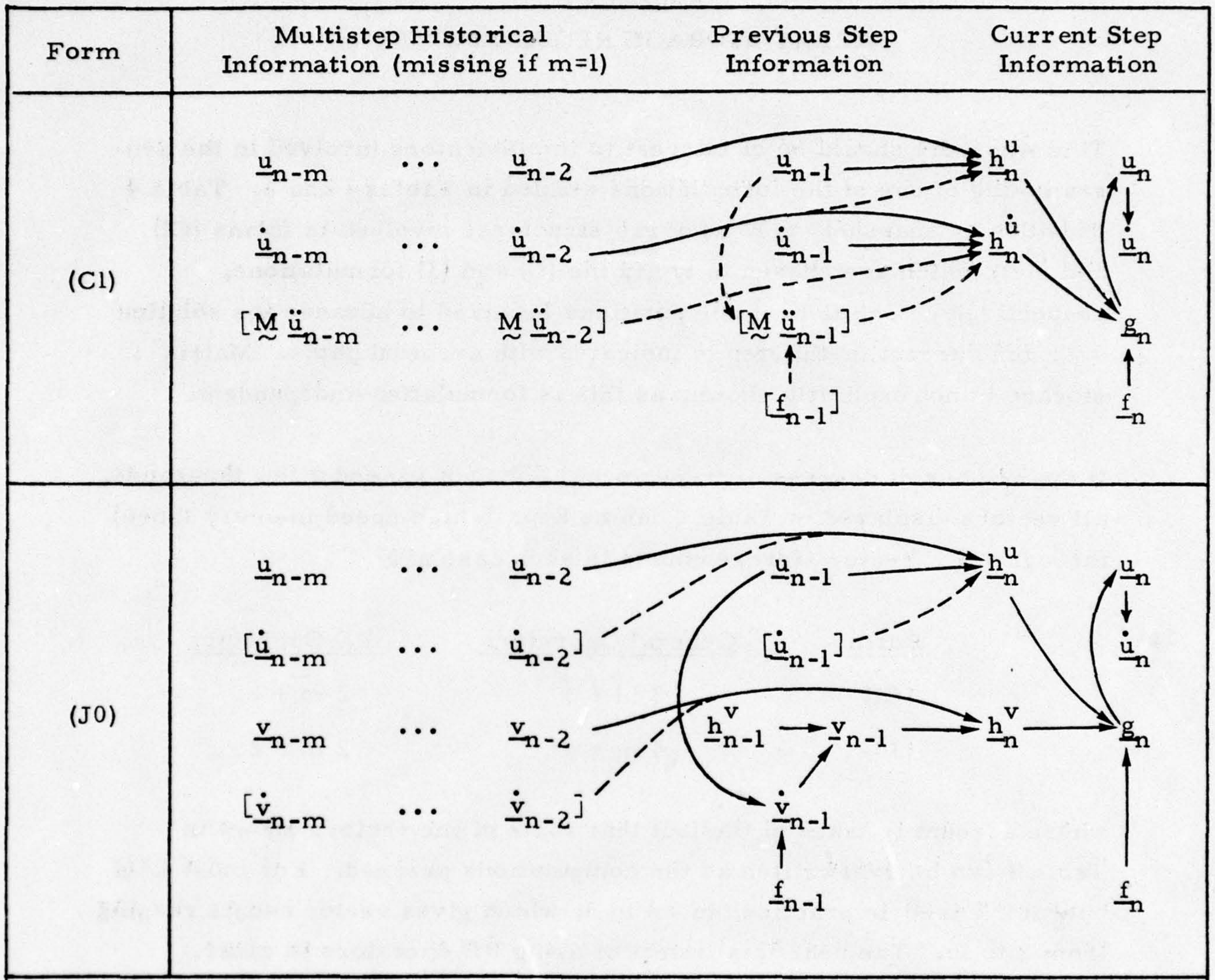
Form	General Operators	BD Operators
(Cl)	3 m + 3	2 m + 1
(J0)	4 m + 4	2 m + 2

where account is taken of the fact that some of the vectors shown in Table 4 can be overwritten as the computations proceed. For most LMS operators used in practice, m = 1 to 3, which gives vector counts ranging from 3 to 16. The beneficial effect of using BD operators is clear.

If n_f exceeds a certain value (typically 3000 to 10000), it is necessary to place some or all vectors on auxiliary storage. In this case the minimal number of core vectors becomes 2, one of which serves as a read buffer.

In a dynamic memory management environment, the transient analyzer typically begins by requesting storage for matrix block buffers to the data management system. Remaining core storage is then parsed in n_f-word blocks to accommodate vector structures. The vector storage area

Table 4
VECTOR STORAGE ARRANGEMENT FOR (C1) AND (J0) FORMULATIONS



NOTES:

1. If a BD operator pair ($\underline{b} = \underline{d} = \underline{0}$) is used, information enclosed in brackets is not required for advancing the computations, and those computational steps identified by broken-line paths may be omitted.
2. Storage arrangement for $\underset{\sim}{M}$, $\underset{\sim}{D}$, $\underset{\sim}{K}$, $\underset{\sim}{E}$ is not shown, as matrix storage requirements are formulation-independent.

is commonly administered as a "revolving pool," which can be viewed as a rectangular array of row dimension n_f whose columns are assigned to specific vectors through a pointer string linked to a cyclic reservation scheme.

COMPUTATIONAL ASPECTS OF TIME INTEGRATION PROCEDURES IN STRUCTURAL DYNAMICS

PART II: ERROR PROPAGATION

K. C. Park
C. A. Felippa

January 1977

Structures Laboratory
Lockheed Palo Alto Research Laboratory
Palo Alto, California

ABSTRACT

The propagation of computational error in the direct time integration of the equations of structural dynamics is investigated. Asymptotic error propagation equations corresponding to the computational paths presented in Part I are derived and verified by means of numerical experiments. It is shown that there exists an implementation form that achieves optimum error control when used in conjunction with one-derivative methods. No such form is found for two-derivative methods. A numerical beating phenomenon is observed for certain implementations of the average acceleration method and the trapezoidal rule, which, from an error propagation standpoint, is highly undesirable.

CONTENTS OF PART II

CONTENTS OF PART II
(Continued)

Section 1
INTRODUCTION

The material presented in Part I may be sufficient for those readers who are primarily interested in organizational aspects of the computer implementation of time integration procedures. Part II, on the other hand, is devoted to a detailed analysis of the computational error associated with the calculation sequences described in Part I, and provides substantiation for claims made as regards the error propagation behavior of those implementation forms. This Part endeavors to answer questions such as: which implementation form minimizes error propagation effects?, what are the advantages of one-derivative LMS operators over two-derivative methods?, how do implementation forms affect the algorithmic properties of the integrator?, and, finally, how can the importance of computational errors be assessed in advance of the actual computations? The answers to these questions have obvious bearing on the implementation of advanced error control strategies in time integration packages developed to support general-purpose structural analyzers.

The potential importance of computational error propagation can be appreciated from the following considerations. A transient response analysis by implicit integration procedures typically involves 10^2 to 10^4 time steps. Each step in turn requires a substantial number of algebraic manipulations. Computational errors introduced at each time step are propagated to subsequent time stations by the integration operator. The compound effect of propagation and accumulation of local errors can seriously jeopardize the accuracy of the solution in many instances, especially in large-scale problems.

Organization of Part II

A summary of new results obtained in this Part is provided to aid the reader in progressing through subsequent sections. An introductory section presents basic concepts required in the error analysis. Sources of local computational error are exhibited in some detail, and mechanisms of propagation and accumulation described. Governing equations for the propagation of a single local error source for a linear system are then introduced; it is shown that the study of the propagated error suffices to characterize the behavior of the overall computational error.

Computational error sources are classified into operational and inherent according to the functional dependence of the associated propaged error on the stepsize h. The operational error is introduced by inexactness in the solution of the algebraic equations that must be solved at each step, whereas the inherent error is due to uncertainties in the specification of applied forces.

A linear, undamped oscillator model is used to examine the behavior of the propagated operational error as a function of the sampling rate. An asymptotic analysis technique is used to show that the long term behavior of the propagated operational error can be characterized, for small stepsizes, by a limit differential or integro-differential propagation equation. Examination of the amplification factor associated with these equations indicates that the computational sequence labeled as path (0) in Part I is immune to local error propagation, hence achieving optimal accumulated error behavior.

An important source of local inherent error in nonlinear structural dynamics is exhibited by consideration of a sample nonlinear system treated by the pseudo-force linearization technique. While a complete analysis of the inherent error propagation in nonlinear problems is not undertaken, its qualitative "semilocal" behavior is illustrated through a linearized treatment. A more thorough treatment of the coupling of the propagated inherent error with computed solutions of highly nonlinear problems will be the subject of future investigations.

Numerical experiments to characterize theerror propagation behavior of several commonly used integrator operators have been performed over a five-magnitude range of sampling frequencies ωh, and the results are displayed in the form of amplification factor spectra. In the low-frequency end, these results agree very closely with the predictions of the asymptotic analysis. In the high-frequency end, an interesting error growth is noted for weakly A-stable operators such as the trapezoidal rule (and its two-derivative counterpart, the average acceleration method) under certain computational paths. This phenomenon is shown analytically to be caused by "numerical beating" associated with the asymptotic appearance, as $\omega h \rightarrow \infty$, of secular terms related to closely allied characteristic roots of the propagation equation. Finally, the implications of the preceding results on the organization of problem-adaptive time-integration software are discussed.

Section 2
SUMMARY OF NEW RESULTS IN PART II

The analysis of computational error propagation in linear dynamic problems relies on the decomposition of error sources over a frequency spectrum encompassing sampling rates normally encountered in practice. To our knowledge, this approach has not been previously used for that purpose. More specific results include:

1. Classification of computational error sources according to their importance with respect to the sampling rate.

2. Derivation of deterministic local error propagation equations for the implementation forms discussed in Part I.

3. Tabulation of accumulated error estimates as function of implementation form and local error correlation properties.

4. Exhibition of a numerical beating phenomenon associated with high-frequency error propagation of certain implementations of the trapezoidal rule and the average acceleration method.

5. Display of experimentally generated spectral error propagation characteristics of specific integrators, and comparison with analytical predictions.

Section 3
COMPUTATIONAL ERRORS

Definition

Let $\underline{u}(t_i)$ denote the solution of the *difference* equations (15) obtainable with *exact* computations. The computational error $\underline{e}(t_i)$ is defined as the residual

$$\underline{e}_i = \underline{e}(t_i) = \tilde{\underline{u}}(t_i) - \underline{u}(t_i) \qquad (29)$$

where $\tilde{\underline{u}}(t_i)$ is the *computed* solution. The computational error results from the propagation and accumulation of local errors committed at each integration step. Consequently, an analysis of (29) must begin with a study of the local errors.

Local Errors

Local errors may be studied *in vacuo* by ignoring, for the moment, the processes of propagation and accumulation. At each time step t_n, we effectively solve the "perturbed" equation

$$(\underset{\sim}{E} + \Delta\underset{\sim}{E})\,(\underline{u}_n + \Delta\underline{u}_n) = \underline{g}_n + \Delta\underline{g}_n \qquad (30)$$

instead of

$$\underset{\sim}{E}\,\underline{u}_n = \underline{g}_n \qquad (15)$$

In Eq. (30), $\Delta\underline{g}_n$ and $\Delta\underline{u}_n$ represent the errors made in the calculation of the right-hand side $\underline{g}_n$ and solution vector $\underline{u}_n$, respectively, and $\Delta\underset{\sim}{E}$ is an equivalent perturbation of the coefficient matrix $\underset{\sim}{E}$ in the spirit of the backward error analysis of Wilkinson [12]. We use the notation

$$\epsilon_u = \frac{\| \Delta\underline{u}_n \|}{\| \underline{u}_n \|} \qquad \epsilon_g = \frac{\| \Delta\underline{g}_n \|}{\| \underline{g}_n \|} \qquad (31)$$

to identify relative local errors, where $\| \cdot \|$ denotes any appropriate vector norm.

Sources of Local Error

The three main sources of local error mentioned in the Introduction to Part I will be examined now in more detail. For the ensuing discussion we refer to an expanded form of (30), which display error terms pertaining to difference sources:

$$\underset{\sim}{E}(\underline{u}_n + \Delta\underline{u}_n^r + \Delta\underline{u}_n^s + \Delta\underline{u}_n^f) = \underline{g}_n + \Delta\underline{g}_n^r + \Delta\underline{g}_n^s + \Delta\underline{g}_n^f \tag{32}$$

in which

$$\begin{aligned} \Delta\underline{g}_n^s &= -\Delta\underset{\sim}{E}(\underline{u}_n + \Delta\underline{u}_n) \approx -\Delta\underset{\sim}{E}\,\underline{u}_n \\ \Delta\underline{g}_n^f &= h_\beta h_\delta \Delta\underline{f}_n \end{aligned} \tag{33}$$

and where other undefined terms are defined in the text below.

Rounding errors caused by the use of finite precision arithmetic contaminate all of the computational steps exhibited in Tables 1 and 2. The aggregate effect of those errors is collected in the right-hand side perturbation $\Delta\underline{g}_n^r$, which induces a solution error $\Delta\underline{u}_n^r$. The term $\Delta\underline{g}_n^r$ is relatively unimportant in most circumstances, with a possible exception: errors introduced in the calculation of accelerations in step (a) of the (C0') form (or of the starter of the (C2) form) if the mass matrix is ill-conditioned.

Solution errors made in solving (15) by direct or iterative techniques are represented by the term $\Delta\underline{u}_n^s$ and the corresponding right-hand side residual $\Delta\underline{g}_n^s$. If a direct solution technique is used, the matrix perturbation $\Delta\underset{\sim}{E}$ results from factorization errors. If an iterative solution procedure is used--as required in nonlinear problems--the term $\Delta\underline{u}_n^s$ is augmented by uncertainties arising from the acceptance of solution iterates that have not attained the full digital significance allowed by the condition of the iteration operator.

Finally, *force evaluation errors* affect the accuracy of the right-hand side through the term $\Delta \underline{g}_n^f$ given by (33b), in which $\Delta \underline{f}_n$ denotes the uncertainty in $\underline{f}_n$. The corresponding solution error term is $\Delta \underline{u}_n^f$.

The relative magnitude of right-hand side and solution errors is related by the condition number of matrix $\underset{\sim}{E}$:

$$\begin{aligned} \epsilon_u &\le C(\underset{\sim}{E})\ \epsilon_g \\ C(\underset{\sim}{E}) &= \| \underset{\sim}{E} \| \, \| \underset{\sim}{E}^{-1} \| \end{aligned} \tag{34}$$

From the error analysis viewpoint, rounding and solution error terms can be lumped together into a component called the "operational error" source. Because of the presence of the h^2 factor in (32b), force errors are treated as a separate component called the "inherent error" source.

Remark. The term "inherent error" was introduced by Henrici [1] to identify errors in the evaluation of the forcing term f(y, t) while integrating the single ODE y' = f(y, t). He then used the term "induced error" to denote the equivalent of rounding and solution errors; we prefer the more suggestive term "operational error."

Error Propagation

The foregoing analysis of local errors would be sufficient for time-independent (static) problems. In a dynamic system, however, local errors committed at t_j are *propagated* to the stream of solutions $\underline{u}(t_k)$, $i \ge j$, by the feedback effect of historic terms. The overall computational error (28) at t_n may be expressed as the *accumulation* of propagated sources:

$$\underline{e}_n = \underline{e}(t_n) = \sum_{i=0}^{n} \underline{G}_n^i \left(\sum_{j=0}^{i} \lambda_{jk}^i \ \Delta u_{jk} \right) \tag{35}$$

where λ_{jk}^i are influence coefficients expressing the propagated error at t_i due to the k-th component Δu_{jk} of the local error $\Delta \underline{u}_j$ at t_j, and $\underline{G}_n^i$ is a vector operator mapping the contribution of the i-th station onto $\underline{e}_n$. For general ODE systems, λ^i_{jk} are nonlinear functions of the solution vector $\underline{u}$. In the case of a linear system, however, the principle of superposition can be invoked to eliminate the nonlinear operator $\underline{G}^i_n$, whence (35) simplifies to

$$\underline{e}_n = \sum_{i=0}^{n} \sum_{j=0}^{i} \lambda^i_{jk} \; \Delta u_{jk} \tag{36}$$

where the λ^i_{jk} are independent of $\underline{u}$, and can be collected, after summing on i, into "propagation matrices" $\underset{\sim}{L}$. For constant stepsize h, these matrices assume a regular pattern illustrated below for the case n = 0, ... 3:

$$\begin{Bmatrix} \underline{e}_o \\ \underline{e}_1 \\ \underline{e}_2 \\ \underline{e}_3 \end{Bmatrix} = \begin{bmatrix} \underset{\sim}{I} & \cdot & \cdot & \cdot \\ \underset{\sim}{L}_1 & \underset{\sim}{I} & \cdot & \cdot \\ \underset{\sim}{L}_2 & \underset{\sim}{L}_1 & \underset{\sim}{I} & \cdot \\ \underset{\sim}{L}_3 & \underset{\sim}{L}_2 & \underset{\sim}{L}_1 & \underset{\sim}{I} \end{bmatrix} \begin{Bmatrix} \Delta\underline{u}_o \\ \Delta\underline{u}_1 \\ \Delta\underline{u}_2 \\ \Delta\underline{u}_3 \end{Bmatrix} \tag{37}$$

where $\underset{\sim}{L}_s$ depends only on the station index difference s = n - j. It is therefore sufficient to investigate the propagation of a typical local error source, which can be taken to be $\Delta\underline{u}_o$ for conveniency. The resulting propagation function $\underline{r}(t_n) = \underset{\sim}{L}_n \, \Delta\underline{u}_o$ can be used to fill up the first column of the supermatrix in (37), and other columns follow by simple shifts. The accumulated error $\underline{e}(t_n)$ can then be calculated by applying statistical assumptions on the occurrence of local errors $\Delta\underline{u}_j$ on the right-hand side of (37).

A detailed, component-by-component calculation of $\underline{e}(t_n)$ is seldom useful (or even possible). All that is needed for practical applications is a rough estimate of the relative error $\| \underline{e} \| / \| \underline{u} \|$ given the local level of accuracy (31) and some gross problem identification parameters. An important step in this regard is to effect a spectral decomposition of the propagated error by passing to normal coordinates, and to identify error sources whose frequency spectrum is similar.

Operational and Inherent Error

For reasons outlined above, it is convenient to separate the propagated typical local source into two components:

$$\underline{r}(t_n) = \underset{\sim}{L}_n \Delta \underline{u}_o = \underline{p}(t_n) + \underline{q}(t_n) \qquad (38)$$

where p(t) denotes the operational error resulting from the propagation or rounding and solution errors, and $\underline{q}$(t) is the inherent error, which results from the propagation of force errors. These error functions satisfy the linear propagation difference equations

$$\left.\begin{aligned} \underset{\sim}{E}\, \underline{p}_n &= \underline{g}_n^p \\ \underset{\sim}{E}\, \underline{q}_n &= \underline{g}_n^q \end{aligned}\right. \qquad n = 0, 1, \ldots \qquad (39)$$

under initial conditions discussed in following sections. The right-hand side terms $\underline{g}_n^p$ and $\underline{g}_n^q$ are to be evaluated following the computational sequences described in Part I, in which the external force term $\underline{f}_n$ is set to zero.

The distinction between operational and inherent error arises naturally from the study of their dependence on the stepsize h, or, more precisely, the circular sampling frequency $\Omega = \omega h$. Broadly speaking, the operational error is dominant for "small" stepsizes whereas the inherent error becomes important for "intermediate" stepsizes. The concept of stepsize magnitude is defined more precisely in the following subsection.

Modal Decomposition

As noted in Part I, the transient response of a linear dynamic system can be viewed as the superposition of normal response components of (generally complex) frequencies ω_k, which are the solutions of the eigensystem (26). Given a specific frequency value ω, a stepsize h will be called

Small, if $|\omega h| < 0.1$
Intermediate, if $0.1 < |\omega h| < 2$
Large, if $|\omega h| > 2$ (Nyquist frequency)

In the analysis of stiff-oscillatory differential systems by implicit integration procedures, the stepsize h is generally such that there are normal components in the three foregoing categories. Generally speaking, small stepsizes are used for components to be accurately traced, intermediate stepsizes for components to be roughly traced, and large stepsizes for components to be "filtered out" by the integrator.

All error vectors--local, propagated and accumulated--can be decomposed into normal components. This spectral decomposition technique will be used in following sections to establish practical estimates on the peak amplifications of the propagated operational and inherent errors.

Section 4
THE OPERATIONAL ERROR PROPAGATION PROCESS

The operational error is dominant for those modal response components associated with small stepsizes. Consequently, this is the only error source that the analyst should be concerned about in problems where the bulk of the excitation energy is contained in low frequency modes.

As noted in the preceding section, it is sufficient to study the propagation of a single operational error source at t = 0, and the resulting propagation function $\underline{p}(t)$ can be projected on the normal coordinates:

$$\underline{p}(t) = \sum_k p_k(t)\, \underline{x}_k \qquad (40)$$

The principal features of a typical modal propagation function $p(t) \equiv p_k(t)$ associated with a small stepsize, i.e., $|\omega_k h| < 0.1$, are illustrated in Figure 2 for the undamped case. Three stages may be noted: an initial jump due to local error initial conditions, a transient (boundary layer) period, and a free oscillation regime.

Initial Conditions

The appropriate initial conditions for the modal analog of (39a) are

$$\begin{aligned} p_o &= \Delta u_o^r + \Delta u_o^s \\ \dot{p}_o &= p_o/h_\beta \\ p_{-1} &= p_{-2} = \cdots = 0 \end{aligned} \qquad (41)$$

where the value of $\dot{p}_o$ results from (12a), in which $h_o^p = 0$ on account of (41c). The conditions (41) may be interpreted as the application of a step "error velocity" $\dot{p}_o$ at $t = -h_\beta$, as depicted in Figure 2.

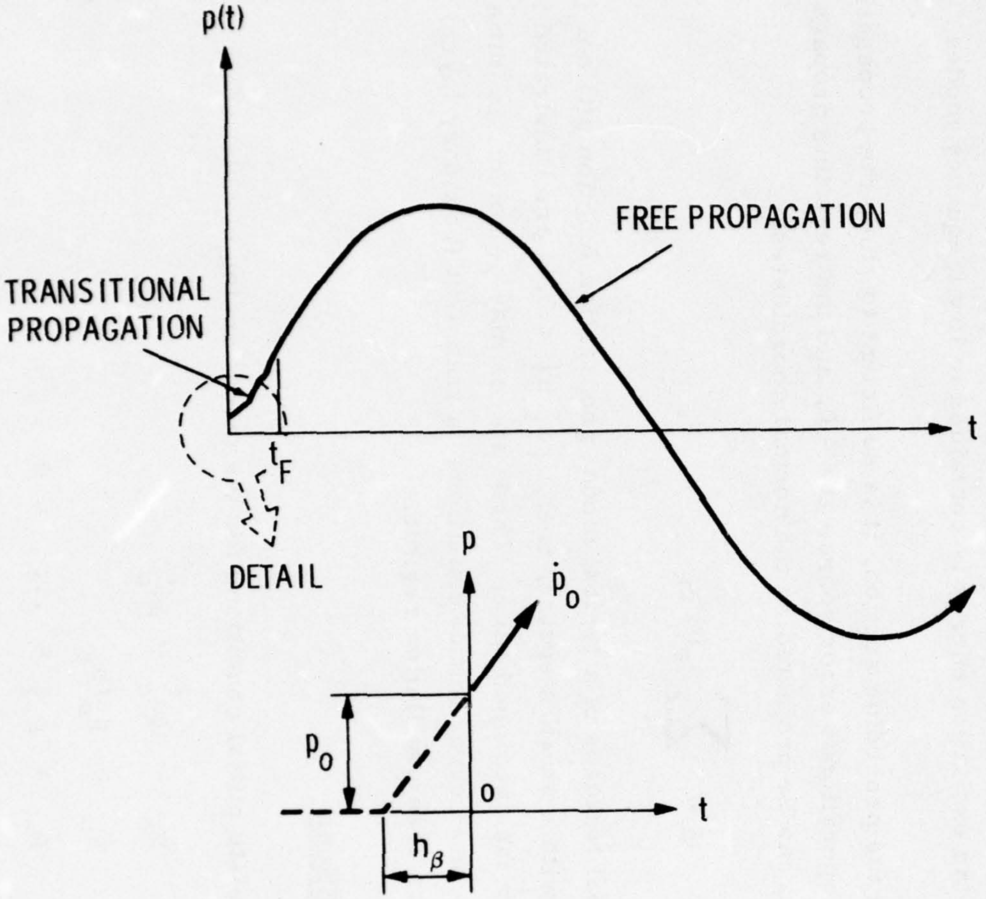


Fig. 2 Schematic propagation history of a small-stepsize operational error source

Transitional Propagation (TP) Period

The TP period extends from $t = 0$ to $t = t_F$, and "bridges" the initial discontinuities (40) with the free propagation regime. The end of this period may be conventionally characterized by an almost uniform velocity $\dot{p}_F = \dot{p}(t_F)$ not varying by more than 1% of $\dot{p}_o$, if ωh is set to zero. If a one-step method is used, or if path (0) is followed, this period lasts one step, i. e., $t_F = h$. On the other hand, for multiple step methods ($m \geq 2$) implemented using paths (1) or (2), t_F depends on the effect of the initial conditions (41) on the spurious roots of the integrator, and on the rapidity of decay of such spurious oscillations; nonetheless, $t_F \approx 2\,m\,h$ represents a good approximation for most commonly used A-stable operators.

The exit conditions from this period have the small-h expansions

$$\left.\begin{aligned} \dot{p}_F &= \rho\, p_o\, \Omega^{-1} + O(1) \\ p_F &= \sigma\, p_o + O(\Omega) \\ \overset{*}{p}_F &= \zeta\, p_o\, \Omega + O(\Omega^2) \end{aligned}\right\} \quad \Omega = \omega h \qquad (42)$$

in which $\overset{*}{p}_n$ denotes the integral $\int_o^n p(t)\,dt$, and where ρ, σ and ζ are scalars of order unity. The values of ρ and σ, which recur in the asymptotic analysis of next section, are tabulated in Appendix C for several commonly used integrators.

Free Propagation (FP) Regime

For $t \geq t_F$, the propagated modal error oscillates with a period associated with the (generally integrator-distorted) frequency pertaining to that mode. The amplitude of the oscillation is primarily determined by the exit conditions (42) and by the sampling frequency $\Omega = \omega h$. The algorithmic properties of the integrator intervene only through second-order effects such as frequency distortion and numerical damping, which vanish rapidly as Ω approaches zero.

The error amplification factor is defined as the ratio

$$\varkappa_p = p_{max}/p_o \tag{43}$$

where p_{max} is the largest absolute value attained by p(t). It is intuitively clear from Figure 2 that for small ωh the maximum of p(t) is reached at the first amplitude peak in the FP regime, as succeeding maxima are abated through numerical damping effects.

Amplification Analysis Procedure

The central problem in the study of operational error propagation consists of establishing the dependence of the error amplification factor (43) on the sampling frequency for specific operators and implementation procedures. A comprehensive study spanning a wide ωh range requires the numerical integration of the governing difference equation (39a) on the computer. This is in fact the approach followed in the numerical experiments reported later. Valuable insight on the effect of the computational path on (43) for $\omega h << 1$ can be obtained, however, by an asymptotic technique.

For a modal component of frequency ω, the dependence of $\varkappa_p$ on the stepsize can be expressed as the Laurent series in $\Omega = \omega h$:

$$\varkappa_p = c_{-1}\,\Omega^{-1} + c_o + c_1\,\Omega + \dots \tag{44}$$

where c_{-1}, c_o, ... are weighting coefficients depending on the integrator and computational path used. The purpose of the asymptotic analysis carried out in the following section is to determine the leading (first non-vanishing) term in (44). The analysis involves two stages:

1. The calculation of the TP exit conditions (42) by solving the propagation difference equations (39) up to t_F. The leading expansion terms in (42) can be obtained by setting $h = 0$ ab initio, which simplifies the computations considerably. Results for specific operators and computational paths are collected in Appendix C.

2. The asymptotic behavior of p(t) in the FP region ($t \geq t_F$) is established by replacing the propagation difference equation (39a)

by the differential (or integro-differential) form approached as ωh tends to zero. A closed form solution matching the TP exit values (42) as initial conditions at t_F then yields the desired information on peak values.

The algebraic manipulations are simplified by ignoring the damping term $\underset{\sim}{D}\,\dot{\underline{u}}$ consistently throughout the error analysis, and working with the homogeneous system

$$\underset{\sim}{M}\,\ddot{\underline{p}} + \underset{\sim}{K}\,\underline{p} = \underline{0} \tag{45}$$

or its modal counterpart

$$\ddot{p} + \omega^2 p = 0 \tag{46}$$

as source differential systems for the propagation difference equation (39a). The introduction of light modal damping in (46) in fact trims the amplification factor by an amount of the order of the first neglected term in (44).

Section 5

ASYMPTOTIC ANALYSIS OF OPERATIONAL ERROR PROPAGATION

The variation of the propagated operational error in the FP regime can be studied as an undamped, free-vibration initial-value problem. In this section we examine the limit differential forms associated with computational paths (2) and (0), as typified by the (C2) and(J0) formulations, respectively.

(C2) Form

The error propagation equations are obtained by specializing the calculations of Table 1 to Eq. (45):

$$(\underset{\sim}{M} + h_\beta h_\delta \underset{\sim}{K})\, \underline{p}_n = \underset{\sim}{M}\, (\underline{h}^p + h_\delta\, \underline{h}^{\dot{p}}) \tag{47}$$

or

$$\underset{\sim}{M}\ \frac{\underline{p}_n - \underline{h}_n^p - h_\beta\, \underline{h}_n^{\dot{p}}}{h_\beta\, h_\delta} + \underset{\sim}{K}\, \underline{p}_n = \underset{\sim}{M}\, \ddot{\underline{p}}_n + \underset{\sim}{K}\, \underline{p}_n = \underline{0} \tag{48}$$

where the vector expression postmultiplying $\underset{\sim}{M}$ has been recognized to be the difference approximation used for the accelerations in the (C2) form. Consequently, as $\omega h \rightarrow 0$, (48) approaches the differential equation (45). Its general solution is easily expressed by passing to normal coordinates. The solution of the associated modal equation (46) with the initial conditions p_F, $\dot{p}_F$ at $t = t_F$, is

$$p(t) = p_F \cos \omega (t - t_F) + \omega^{-1} \dot{p}_F \sin \omega (t - t_F) \tag{49}$$

Substituting the expansions (42a, b) into (49) and estimating the peak value gives

$$\kappa_p = \rho\, (\omega h)^{-1} + O(1) \tag{50}$$

(J0) Form

The analysis of this form is more involved. The error propagation equations follow by specializing (25) to the undamped case:

$$(\underset{\sim}{M} + h_{\beta} h_{\delta} \underset{\sim}{K})\ \underline{p}_n = \underset{\sim}{M}\, \underline{h}_n^{p} + h_{\delta}\, \underline{h}_n^{v} \qquad (51)$$

in which $\underline{v} = \underset{\sim}{M}\, \dot{\underline{p}}$ denotes here the auxiliary vector (24a) associated with $\underline{p}(t)$. This vector is determined recursively from the differential equation, i. e.,

$$\begin{aligned} \dot{\underline{v}}_n &= - \underset{\sim}{K}\, \underline{p}_n \\ \underline{v}_n &= \underline{h}_n^{v} + h_{\delta}\, \dot{\underline{v}}_n \end{aligned} \qquad (52)$$

To eliminate $\underline{v}$ from (51), the historical vector $\underline{h}_n^{v}$ has to be expressed in terms of the value of $\underline{p}$ at past stations. It can be shown that

$$\underline{h}_n^{v} = - h\, \underset{\sim}{K} \sum_{i=-\infty}^{n} \mu_i\, \underline{p}_i \qquad (53)$$

in which μ_i are coefficients resulting from the Taylor expansion of the generating rational function

$$\frac{\delta_o + \delta_{\gamma} z + \ldots\ \delta_m z^m}{\gamma_o + \gamma_1 z + \ldots\ \gamma_m z^m} = \mu_o + \mu_1 + \ldots + \mu_{n-1} z^{n-1} \qquad (54)$$

where γ_i and δ_i are coefficients of the $\underline{v}$-integrator (7b). The $-\infty$ limit in (53) can be replaced by zero on account of (41c). Define now

$$\overset{*}{\underline{p}}_n = h_{\delta}\, \underline{p}_n - h_{\delta}\, \underset{\sim}{K}^{-1}\, \underline{h}_n^{v} / h_{\beta} \qquad (55)$$

It turns out that $\overset{*}{\underline{p}}_n$ is a discrete integral of $\underline{p}(t)$. This can be verified by expressing (9) for $\underline{p}_n$ and $\overset{*}{\underline{p}}_n$

$$\overset{*}{\underline{p}}_n = h_{\delta}\, \underline{p}_n + h\, \underline{d}_n^{p} - \underline{c}_n^{\overset{*}{p}} \qquad (56)$$

in which $\underline{c}_n^{\overset{*}{p}}$ and $\underline{d}_n^p$ can be recursively calculated in terms of p_i using (52). As an example, for the trapezoidal rule ($\gamma_1 = -1$, $\delta_o = \delta_1 = 1/2$), we obtain

$$\overset{*}{\underline{p}}_n = 1/2\, h\, (\underline{p}_o + 2\, \underline{p}_1 + \dots\dots + 2\, \underline{p}_{n-1} + \underline{p}_n) \tag{57}$$

which is the well-known trapezoidal quadrature formula. Inserting (53) into (51) produces the difference expression

$$\begin{gathered} \underset{\sim}{M}\, (\underline{p}_n - \underline{h}_n^p)/h_\beta + \underset{\sim}{K}\, \overset{*}{\underline{p}}_n = \underline{0} \\ \text{or} \\ \underset{\sim}{M}\, \dot{\underline{p}}_n + \underset{\sim}{K}\, \overset{*}{\underline{p}}_n = \underline{0} \end{gathered} \tag{58}$$

where $\dot{\underline{p}}_n$ is the difference expression used in step (b) of Table 2. In the limit $\omega h \rightarrow 0$, (58) becomes the integro-differential equation

$$\underset{\sim}{M}\, \dot{\underline{p}} + \underset{\sim}{K} \int_0^{t_n} \underline{p}\, dt = \underline{0} \tag{59}$$

The solution of the associated normal equation

$$\dot{p} + \omega^2 \int_0^{t_n} p\, dt = 0 \tag{60}$$

satisfying the initial conditions p_F, $\overset{*}{p}_F$ at t_F, is

$$p(t) = p_F \cos \omega\, (t - t_F) - \omega\, \overset{*}{p}_F \sin \omega\, (t - t_F) \tag{61}$$

Introducing (42b, c) into (61) and estimating the peak value produces

$$\varkappa_p = \sigma + O(\omega h) \tag{62}$$

The error amplification factor of the (J0) form is asymptotically independent of the stepsize h. Consequently, this implementation is immune to local propagation error effects.

<u>Remark 1.</u> The choice of the (J) and (C) formulations to exhibit the error propagation behavior of the (0) and (2) paths, respectively, has no special significance. In fact, the same analysis (leading to identical conclusions), could be carried out for an arbitrary auxiliary vector.

Remark 2. An analysis of computational paths (0') and (1) leads to the limit forms (59) and (45), respectively. The corresponding amplification factors have expansions similar to (62) and (50), respectively, but the coefficients ρ and σ are generally different (cf. Appendix C). Consequently, a detailed analysis of these paths will not be given.

Remark 3. The integro-differential equation (59) that governs the error propagation behavior of the (0) path has the important property of being insensitive to the initial step velocity (41b). This smoothing property has been actively exploited in the field of discrete sequential estimation for the design of stochastic filters [13]. The elimination of error amplification effects is in fact analogous to the filtering of process noise through a Kalman filter.

Two-Derivative Methods

An analysis of the operational error propagation for the two-derivative operators mentioned in Appendix Aleads to modal amplification factors of the form (50), regardless of the computational path used. The smoothing effects of the auxiliary vector sequence (52) are lost in the (0) path because computed solutions are independent of the velocities in the undamped case. Consequently, the error filtering effect of the (C0) and (J0) implementations is lost if a two-derivative operator is used. This can be a significant drawback of two-derivative methods when the operational error can severely affect the quality of the answers.

Section 6
ACCUMULATED OPERATIONAL ERROR

The accumulated operational error, denoted by $\underline{e}_n^p$, results from the composition of propagated operational sources. An estimation of $\underline{e}_n^p$ necessarily requires some assumptions on the statistical distribution of sequential local error occurrences. The reader is referred to Henrici [1, 2] for a detailed exposition of statistical analysis techniques. We are here content to illustrate the accumulation process pertaining to a small-stepsize modal component ($\omega h < 0.1$) in the simplest instance where local modal errors are perfectly correlated, i. e.,

$$\Delta u = \Delta u_i = \Delta u_i^r + \Delta u_i^s \;, \quad \text{for each } i \tag{63}$$

For computational paths (1) and (2), the dominant term in the propagation function (49) is, for small $\Omega = \omega h$:

$$p(t) \approx \omega^{-1} \dot{p}_F \sin \omega (t - t_i)$$

Consequently,

$$e_n^p = \omega^{-1} \dot{p}_F \sum_{i=0}^{n} \sin \omega (t_n - t_i) = \omega^{-1} \dot{p}_F \sum_{j=0}^{n} \sin j\Omega$$

$$= \omega^{-1} \dot{p}_F \sin 1/2(n-1)\Omega \;\; \sin 1/2\,\Omega \;\; \operatorname{cosec} 1/2\,\Omega \tag{64}$$

Expressing $\dot{p}_F$ in terms of (63) through (42a), the following expansion for the peak value of e_n^p is obtained:

$$e_{max}^p = \max_n |e_n^p| = \rho\, \Omega^{-2} \Delta u + O(\Omega^{-1}) \tag{65}$$

A similar argument for path (0), using the dominant term in (61), yields

$$e_{max}^p = \sigma\, \Omega^{-1} \Delta u + O(1) \tag{66}$$

Leading terms of the asymptotic expansions of the propagated and accumulated errors in the low and high frequency regions of the modal spectrum are collected in Table 5. The error exponent ν defined there is a measure of the correlation of sequential local error occurrences.

Remark. In single ODE's, where rounding is the only significant local error source, it is generally assumed that sequential local errors are normally distributed about a zero mean, which leads to a favorable exponent $\nu = 1/2$. For large-scale stiff systems, however, that assumption may be unrealistic. If the system is at least moderately ill-conditioned, the solution error term $\Delta \underset{\sim}{E}\, \underline{u}_n$ tends to dominate over the rounding error. As the same $\underset{\sim}{E}$ is normally used for many steps, the factorization error perturbation $\Delta \underset{\sim}{E}$ remains unchanged; furthermore, the low-frequency modal components in $\underline{u}_n$ vary slowly at the usual sampling rates. The net effect is that successive local errors may be highly correlated, and the error exponent approaches the most unfavorable value $\nu = 1$.

Table 5

ORDER OF LEADING TERM IN ASYMPTOTIC EXPRESSIONS OF MODAL OPERATIONAL ERROR

Integrator Type	Comp. Path	Small-stepsize, low-frequency range $\Omega << 1$		Large-stepsize, high-frequency range $\Omega >> 1$			
				Strongly A-stable operators		Weakly A-stable operators	
		POE	AOE	POE	AOE	POE	AOE
Pair of one-derivative operators (7)	(0')	$O(1)$	$O(\Omega^{-\nu})$	$O(1)$	$O(\Omega^{\nu})$	$O(\Omega)$	$O(\Omega^{1+\nu})$
	(0)					$O(1)$	$O(\Omega^{\nu})$
	(1, 2)	$O(\Omega^{-1})$	$O(\Omega^{-1-\nu})$				
Two-derivative operator (Alb) (conventional implementation)	(0', 1)	$O(\Omega^{-1})$	$O(\Omega^{-1-\nu})$	$O(1)$	$O(\Omega^{\nu})$	$O(\Omega)$	$O(\Omega^{1+\nu})$
	(2)					$O(1)$	$O(\Omega^{\nu})$

NOTES:

1. POE: Modal amplification factor (43) of propagated operational error.
2. AOE: Modal amplification factor $e^p_{max}/\Delta\bar{u}$ of accumulated operational error where $\Delta\bar{u}$ is an statistical average of local modal errors $|\Delta u_i|$.
3. The error exponent ν varies from 1 for perfectly correlated sequential local errors to 1/2 for uncorrelated (random) sequential local errors. (The value appropriate to the low-frequency range is not necessarily equal to the value appropriate to the high-frequency range, but virtually nothing is presently known as to the dependence of ν on Ω.)
4. A strongly (weakly) A-stable operator is characterized by the largest root of the polynomial $\beta_o z^m + \ldots + \beta_m$ having a modulus less than (equal to) unity. This root characterizes the filtering properties of the operator as regards suppression of high-frequency noise [4].

Section 7
INHERENT ERROR

A Sample Nonlinear Equation

As previously noted, the inherent error arises from uncertainties in the forcing term. As a general rule, the effect of inaccuracies in the specification of external forces can be ignored in practice, unless the external force field results from the concurrent solution of a coupled initial value problem (e.g., fluid-structure interaction, coupled thermoelasticity). Internal force errors, on the other hand, may be important in nonlinear structural dynamics, as the following sample problem illustrates. Consider an undamped nonlinear dynamic system governed by the state equilibrium equation

$$\underset{\sim}{M}\,\underline{\ddot{u}} + \underline{N}(\underline{u}) = \underline{f}(t) \tag{67}$$

where $\underline{N}(\underline{u})$ is a nonlinear stiffness term. Two widely used linearization techniques for treating that term are

$$\begin{aligned} \underline{N}(\underline{u}_n) &= \underline{N}(\underline{u}_{n-1}) + \underset{\sim}{K}_{n-1}(\underline{u}_n - \underline{u}_{n-1}) - \Delta\underline{N}_n \\ \underline{N}(\underline{u}_n) &= \underset{\sim}{K}\,\underline{u}_n - \underline{Q}_n = \underset{\sim}{K}\,\underline{u}_n - (1+\varphi)\,\underline{Q}_{n-1} + \varphi\,\underline{Q}_{n-2} \cdot \Delta\underline{N}_n \end{aligned} \tag{68}$$

where $\underset{\sim}{K}_{n-1}$ is the Jacobian of $\underline{N}(\underline{u})$ evaluated at $\underline{u}_{n-1}$, $\underset{\sim}{K}$ is a Jacobian evaluated at the reference state $\underline{u} = \underline{0}$, $\underline{Q}$ is a nonlinear "pseudo-force" term, φ is a extrapolation parameter in the range 0 to 1, and $\Delta\underline{N}_n$ is the local approximation error. The Taylor series linearization (68a) is called the tangent-stiffness method, whereas the fixed-point linearization (68b) is referred to as the pseudo-force or secant-stiffness method in the engineering literature. For the sake of brevity, we use the latter technique in the following analysis. Substitution of (68b) into (67) evaluated at t_n yields

$$\underset{\sim}{M}\,\underline{\ddot{u}}_n + \underset{\sim}{K}\,\underline{u}_n = \underline{\hat{f}}_n + \Delta\underline{\hat{f}}_n \tag{69}$$

where the effective force $\hat{\underline{f}}_n$ and inherent error source $\Delta\hat{\underline{f}}_n$ are given by

$$\begin{aligned} \hat{\underline{f}}_n &= \underline{f}_n + (1+\varphi)\,\underline{Q}_{n-1} - \varphi\,\underline{Q}_{n-2} \\ \Delta\hat{\underline{f}}_n &= \Delta\underline{f}_n + \Delta\underline{N}_n \end{aligned} \qquad (70)$$

In (70b), $\Delta\underline{f}_n$ denotes the applied or external force error, as in (33b). The practical importance of the term $\Delta\underline{N}_n$ should not be overlooked. In problems involving path-dependent material nonlinearities, such as plasticity, the extrapolation error in (68b) may lead to a pseudo-force error of the order of 1-10% of $\underline{f}_n$ unless a reliable equilibrium-correcting strategy is used.

In a highly nonlinear problem, the inherent error source (70b) propagates in a complex manner. As the modal spectrum changes with time, the principle of superposition of individual sources is not applicable. The following considerations are restricted to linear systems, but should also provide insight into the "semilocal" force-error propagation in mildly nonlinear problems.

Linearized Analysis

Eq. (69) is linearized by assuming that the right-hand side terms are independent of $\underline{u}$. The propagated inherent error $\underline{q}(t)$ associated with a single source $\Delta\underline{f}_o$ at t = 0 then satisfies the linear difference equation (39b). The appropriate initial conditions are

$$\begin{aligned} \underline{g}_o^q &= h_\beta\, h_\delta\, \Delta\underline{f}_o \\ \underline{q}_o &= \underset{\sim}{E}^{-1}\, \underline{g}_o^q \\ \dot{\underline{q}}_o &= \underline{q}_o / h_\beta \\ \underline{q}_{-1} &= \underline{q}_{-2} = \cdots = \underline{0} \end{aligned} \qquad (71)$$

All of the results obtained in previous sections concerning operational error propagation apply to the inherent error function $\underline{q}(t)$ if $\underline{p}_o$ and $\dot{\underline{p}}_o$ are replaced by $\underline{q}_o$ and $\dot{\underline{q}}_o$ given by (71b, c). It is not convenient, however, to use the analog q_{max}/q_o of (43) as modal amplification factor. A more significant measure is the ratio of modal force errors:

$$\varkappa_q = \frac{\omega^2 q_{max}}{\Delta f_o} = \frac{\Omega^2}{1 + \Omega^2} \quad \frac{q_{max}}{q_o} \tag{72}$$

which agrees with q_{max}/q_o in the high-frequency end $\Omega = \omega h >> 1$.

Inherent Error of the Average Acceleration Method

The average acceleration (AA) method, which is a member of the two-derivative Newmark's operator family [6], is presently the most commonly used implicit integration scheme in linear structural dynamics. It is also widely used in nonlinear dynamics notwithstanding its poor performance as regards high-frequency noise amplification, experimentally noted by many investigators [9, 14]. The error growth is caused by the coalescence of the two characteristics roots of the integration operator as $\Omega = \omega h \to \infty$ if either computational path (0) or (1) is followed. Near-root coalescence brings out a "numerical beating" phenomenon, which is briefly analyzed below and substantiated in the numerical experiments.

The propagation difference equation (39b) associated with the AA method is (cf. Appendix C):

$$q_n - 2 q_{n-1} + q_{n-2} + 1/2 \, \Omega^2 (q_n + 2 q_{n-1} + q_{n-2}) = 0 \qquad (73$$

The general solution of (73) is

$$q_n = C_1 \lambda_1^n + C_2 \lambda_2^n \tag{74}$$

where

$$\lambda_{1,2} = e^{\pm i \emptyset} \, , \quad \emptyset = \tan^{-1} \frac{\Omega}{1 - 1/4 \, \Omega^2} \tag{75}$$

The appropriate initial conditions for (73) are $q_{-1} = 0$, $q_o = q_o$, whence (74) becomes

$$q_n = q_o \, (\lambda_1^{n+1} - \lambda_2^{n+1}) \, / \, (\lambda_1 - \lambda_2) \tag{76}$$

which, upon introducing (75), can be reduced to

$$q_n = q_o \, \Omega^{-1} (1 + 1/4 \, \Omega^2) \sin n \phi \tag{77}$$

The beating frequency ϕ, given by (75), approaches $4 \, \Omega^{-1}$ if $\Omega >> 1$. The amplification factor (72) associated with (77) is

$$\kappa_q = \Omega (1 + 1/4 \, \Omega^2)/(1 + \Omega^2) \approx 1/4 \, \Omega \text{ if } \Omega >> 1 \tag{78}$$

The same beating phenomenon occurs in the use of the trapezoidal rule when implemented in the (C0') form. In this case, the appropriate initial conditions are $q_{-1} = -q_o$ (since $h_\beta = 1/2h$), which causes the propagation function (77) to double.

Identical conclusions apply to the propagation of operational error. However, this source is not usually important in the high-frequency range.

Section 8
NUMERICAL EXPERIMENTS

In formal paper presentation, numerical results, if any, are generally offered as verification of the correctness of the analytical derivations. The implication is that the analytical framework was somewhat put together magically by the investigators, who then walk to the computer to reap the deserved confirmation of their brain labors. As detailed in Appendix D, however, computationally-oriented research seldom conforms to such a neatly staged schedule. In real life, numerical experiments and theory development proceed concurrently, with the former often suggesting avenues of progress for the latter. The synergistic interaction of experiment and theory has in fact been largely responsible for the success of the present study. With these sobering thoughts out of the way, we proceed to describe summarily the procedures followed in the experimental phase; this is followed by a detailed discussion of the numerical results presented in Figures 3 through 27.

Experimental Setup

The numerical results presented here were obtained by solving the propagation difference equations (39), specialized to an undamped linear oscillator, at a prescribed set of sampling frequencies ranging from $\Omega = \omega h = 10^{-2}$ to 10^{3}. The study parameters considered were: (a) the integration method, (b) the error source type (operational/inherent), and (c) the computational path followed. Observed peak responses were collected into amplification factor arrays for subsequent generation of amplification spectrum diagrams. In addition, options were provided to produce time history plots of the propagation process at specific sampling frequencies.

The computer program that carries out those tasks was not assembled as a final product, but evolved in stages closely integrated with progress in the theoretical understanding of the error propagation process. The initial version was used to investigate the error propagation characteristics of one-step, one-derivative methods under the basic computational paths (0-2).

Multiple step one-derivative operators were incorporated once the basic properties of one-step operators were reasonably well understood, and, finally, a set of two-derivative operators was implemented. The distinction between paths (0) and (0') was detected in the final stages of this study (cf. Appendix D).

The present version contains eighteen one- and two-derivative operators as built-in methods. These integrators are simply accessed by symbolic key, e. g., 'METHOD TR' calls for the trapezoidal rule coefficients. (Some of the built-in operators are actually integrator families, such as Newmark's and Wilson's, in which case the identification key is accompanied by one or more parameter values.) The program also allows the direct specification of integrator coefficients.

The error analysis code can be operated in either batch or conversational-interactive mode through a free-field, problem-oriented language. The interactive capability has been found to be valuable when the program is used as an experimental tool to guide and/or confirm theoretical developments. Results can be displayed in either alphanumeric or graphic mode. Graphic display is generated through the DISSPLA plotting package. Graphic output produced in interactive work can be directly displayed on in-line Tektronic terminals. These "quickie" plots are useful for program and input data debugging, and also represent a convenient means of preparing the generation of high quality off-line plots.

Organization of Numerical Results

A comprehensive set of numerical results on error propagation characteristics of specific integration methods is presented in Figures 3 to 27. Two set-inclusion criteria were used: (a) the result pertains to a commonly used integration method, or (b) the result illustrates a particularly important or noteworthy feature previously discussed in this part. All of the plots were entirely computer-produced on the Comp80 microfilm recorder through the DISSPLA plotting package; only the figure captions have been added.

The plot series is organized as follows.

Figures 3-11: Amplification spectra of propagated operational error for one-derivative methods

Figures 12-17: Amplification spectra of propagated operational error for two-derivative methods

Figures 18-19: Time history plots illustrating the propagation of small- and large-stepsize operational error sources

Figures 20-22: Amplification spectra of propagated inherent error for one-derivative methods

Figures 23-24: Amplification spectra of propagated inherent error for two-derivative methods

Figures 25-27: Time history plots of propagation of inherent error sources illustrating the formation of "beats" in the average acceleration method

In all results pertaining to one-derivative methods, the same operator was used in the discretization of $\underline{u}$ and $\underline{v}$.

Operational Error Propagation for One-Derivative Methods

Figures 3 through 6 show amplification factors corresponding to the general one-derivative, one-step A-stable integrator family

$$\underline{u}_n = \underline{u}_{n-1} + h\,[f\,\dot{\underline{u}}_n + (1 - f)\,\dot{\underline{u}}_{n-1}] \;,\quad 1 \geq f \geq 1/2 \tag{79}$$

for the values f = 1.0, 0.60, 0.55 and 0.50. The limit cases f = 1 and f = 0.5 correspond to the backward Euler method and trapezoidal rule, respectively. These four plots provide a fairly good picture of the transition from a highly numerically-damped BD operator (backward Euler) to a weakly A-stable operator (trapezoidal rule). The gradual development of resonant amplification in the large stepsize (high frequency) range $\Omega > 2$ for path (0') is especially noteworthy.

Figures 7 to 8 illustrate two-step methods. Gear's two-step operator, being an A-stable, highly damped BD scheme, displays amplification behavior similar to that of backward Euler. Figure 8 shows results for the two-step member (27b) of Park's three-step starting family (27).

Figures 9 to 11 illustrate three-step methods. The unbounded amplification of Gear's three-step method (Figure 9) for $1/2 < \Omega < 2$ is typical of a stiffly stable method, for which the instability region intercepts a portion of the imaginary frequency axis. Figure 10 shows results for a method labeled as Formula 3 in Jensen's paper [15]; the original stiffly-stable method has been A-stabilized by addition of recommended numerical damping terms. This operator is interesting in that it is the only one-derivative method for which path (1) has been found to be inferior to path (2) in this experimental series. Park's three-step method, shown in Figure 11, displays amplification characteristics similar to those of the other two A-stable BD operators (Figures 3 and 7) despite having less numerical damping.

In all cases, an excellent agreement with the predictions of the asymptotic analysis expansions (51) and (62) can be observed in the small stepsize region $\Omega < 0.1$. In fact, the agreement is surprisingly good even in the intermediate stepsize region and up to $\Omega \sim 1$ for most methods.

Operational Error Propagation for Two-Derivative Methods

Error amplification plots associated with two-derivative implicit operators show at most two curves corresponding to paths (0) and (2), as paths (0), (0'), and (1) coincide. For BD operators such as Houbolt's, all paths are identical.

Figure 12 shows amplification spectra for the average acceleration method. This operator deserves special attention because it is the most widely used integrator in linear dynamics. Moreover, this scheme is amenable to an exact amplification analysis, typified by Eqs. (73) to (77), which provides factors valid for all sampling frequencies. The curve for path (0) has the equation

$$\varkappa_p = \Omega^{-1} (1 + 1/4\, \Omega^2) \qquad (80)$$

which is symmetric about the Nyquist frequency $\Omega = 2$.

The central difference method (Figure 13) is the only explicit operator included in this series on account of its wide application in shock dynamics. (The integration of the propagation equations was carried out through a special branch in the program, and the label 'path (0)' has in fact no meaning.) It is interesting to note the smallness of the amplification factor even while very close to the folding frequency $\Omega = 2$. As in the case of the average acceleration method, an exact treatment of the propagation difference equations presents no difficulties and leads to the expression

$$\varkappa_p = \Omega^{-1} (1 - 1/4\,\Omega^2)^{-1/2} \quad , \text{ for } \quad \Omega < 2 \qquad (81)$$

The Houbolt method (Figure 14) displays the excellent high-frequency noise suppression behavior that can be expected from a BD operator with high numerical damping. This feature has contributed to the popularity of this integrator in nonlinear dynamics.

The linear acceleration method (Figure 15) is a conditionally stable member of two operator families: Newmark's (for $\beta = 1/6$) and Wilson's (for $\theta = 1$). The amplification factor becomes infinity at the folding frequency $\Omega = 12^{1/2} = 3.464$. It is interesting how much worse path (2) is for this particular method.

Wilson's one-parameter operator family is A-stable for $\theta > 1.3660$. Figures 16 and 17 illustrate the cases $\theta = 1.37$ and $\theta = 2.0$, respectively. This integrator has been criticized on account of its poor error propagation characteristics when implemented as a self-starting method [16]. In the usual implementation, the solution is advanced beyond the next time station (t_n) up to $t_n + (\theta - 1)h$ by an extrapolation procedure; the method then regresses to t_n through interpolation. High error amplifications (e.g., $\varkappa_p \sim 10^4$ at $\Omega \sim 0.1$) are in fact common to all of the so-called "advanced" LMS methods. In our study, the Wilson operator was implemented as a regular three-step LMS method [11]. Figures 16 and 17 indicate that in such a case the low-frequency error amplification behavior is similar to that of other two-derivative methods, whereas the high-frequency attenuation resembles that of Houbolt's method (cf. Figure 14). The numerical results suggest that the dependence of the small-stepsize amplification factor on the parameter θ is

$$\begin{aligned} \varkappa_p &= \theta\,\Omega^{-1} + O(1) \;, && \text{for paths (0), (0') and (1)} \\ \varkappa_p &= 6\,\Omega^{-1}/\theta^2 + O(1), && \text{for paths (2)} \end{aligned} \qquad (82)$$

If these expressions are correct, all paths would be asymptotically identical for $\theta = 6^{1/3} = 1.817$.

Time History Propagation of Operational Error Sources

Figures 18 and 19 are included to illustrate the temporal propagation of a unit operational error source pertaining to small stepsize ($\Omega = 0.1$) and large stepsize ($\Omega = 100$) normal components, respectively, for the trapezoidal rule. The propagation histories in Figure 18 display the typical smooth harmonic behavior of low frequency components, as predicted by Eqs. (49) and (61). The transitional propagation period (cf. Figure 2) is barely perceptible at the scale of the plot, since it lasts only one step. Figure 19 illustrates the high frequency "numerical beating" behavior associated with path (0'); histories for the other paths have been suppressed as they would be undistinguishable from the time axis.

Inherent Error

The amplification of inherent error sources, as measured by the force error ratio (72), cannot be significant in the small stepsize range on account of the presence of the factor Ω^2. With the exception of weakly A-stable operators implemented under certain computational paths, all methods display bounded amplification as $\Omega = \infty$. Because of this similarity of behavior, only five representative cases are presented.

Figures 20 through 22 show inherent error amplification spectra associated with the backward Euler scheme, Park's three-step method, and the trapezoidal rule, respectively. These plots are chosen to typify the behavior of strongly and weakly A-stable one-derivative methods. The only unbounded growth occurs for the trapezoidal rule if path (0') is used, in which case $\kappa_q \sim 1/2\Omega$ for $\Omega >> 1$.

Figures 23 and 24 depict inherent error amplification spectra of the average acceleration method and of Wilson's operator ($\theta = 1.37$), respectively. The former exhibits unbounded growth for paths (0), (0') and (1), as predicted by Eq. (78).

Finally, Figures 25 through 27 are included to illustrate the process of formation of high frequency "beats" for the average acceleration method. A unit modal force error is introduced at $t = 0$. At a sampling frequency $\Omega = \omega h = 1$ (~ 6 samples/cycle), the propagation histories of Figure 25 still exhibit rough harmonic behavior. The incipient emergency of beats in path (0) at $\Omega = 10$ can be perceived after some study of Figure 26. The beats are perfectly developed at $\Omega = 100$ (Figure 27).

Section 9
APPLICATIONS

We have so far examined the effects of implementation forms on the behavior of the propagated computational error. This error is different from the algorithmic error as commonly assessed in terms of numerical damping and phase distortion. From the foregoing analysis, it is clear that the computational error should be taken into account in the design of reliable transient analysis software. As this aspect has rarely been dealt with in present structural dynamics programs, we sketch here how the results of the error analysis can be put to use in the development of stepsize control strategies. We consider the simplest case of linear structural dynamics, in which the bulk of the excitation energy is associated with low-frequency components, and assume that the (J0) form has been implemented. A rough idea of the peak (relative) computational error can be obtained from (34) and Table 5:

$$\frac{\| \underline{e} \|_{\max}}{\| \underline{u} \|} \sim \sigma \, (\omega_{\min} h)^{-\nu} \, C(\underset{\sim}{E}) \, \varepsilon_g \qquad (83)$$

where $\omega_{\min}$ is an estimate of the lowest frequency. A stepsize h is initially selected, usually on the basis of algorithmic accuracy requirements, and $\underset{\sim}{E}$ is formed. An order-of-magnitude estimate of the condition number $C(\underset{\sim}{E})$ can be economically obtained once $\underset{\sim}{E}$ is factored [12]. (Note that $C(\underset{\sim}{E})$ is a function of h, and varies from $C(\underset{\sim}{M})$ as $h \rightarrow 0$ to $C(\underset{\sim}{K})$ as $h \rightarrow \infty$.) The local accuracy level ε_g can be estimated as $n_f \, 10^{-d}$, where d is the digital significance used. Because of the high uncertainty on most of the quantities in (83)--particularly the error exponent--the frequency $\omega_{\min}$ needs to be estimated only within an order of magnitude. If (83) exceeds a desired accuracy threshold, a higher arithmetic precision level is likely needed. In any case, (83) provides an idea of the minimum acceptable stepsize. If there is a capability for adjustable precision work, a much sharper estimate of (83) can be obtained by carrying out a short integration in two precision modes.

Another application of the error analysis is the problem-adaptive selection of optimal implementation forms. For example, if the mechanical system is undamped, the use of a pair of BD operators in conjunction with the (Cl) form would be desirable in terms of minimizing the computational effort (cf. Table 3). A prior estimation of the computational error determines whether such strategy is permissible; if not, the integration should be carried out with the somewhat more expensive (J0) form. Other situations can be similarly treated.

Section 10
CONCLUSIONS

The material covered in Part II represents an initial contribution to the study of computational error in direct time integration of second-order ODE systems. Although the application to structural dynamics has been emphasized, the conclusions readily extend to other physical problems modeled by stiff-oscillatory systems. A summary of our main findings follows.

Implementation forms are found to have a decisive effect on the error propagation behavior pertaining to low-frequency, small-stepsize normal components, while the algorithmic properties of the integrator play a comparatively minor role. Both factors are equally significant in high-frequency error propagation.

We have shown that the best error control is achieved when one-derivative LMS methods are used in conjunction with the path (0) implementation. In this case, the local error propagation equation approaches, as the stepsize is reduced, an integro-differential form that is insensitive to local velocity perturbations. This favorable behavior cannot be duplicated by any conventional implementation of a two-derivative method.

The trapezoidal rule and the average acceleration method are often considered to be optimal integration operators in linear dynamics due to their algorithmic properties: no artificial damping and minimal phase distortion within the class of A-stable methods. Under certain computational paths, however, these methods exhibit a numerical resonance phenomenon that amplifies high-frequency error components. This effect seriously undermines the viability of such implementations in the frequent cases where an effective filtering of high-frequency noise is required.

It should be stressed that these conclusions are of practical importance only if the computational error propagation can significantly affect the quality of the computed response. This condition is seldom easy to identify before the problem is presented to the computer. It is hoped that the information presented here will be useful for diagnosing whether accuracy problems due to error propagation exist, and, if so, in providing realistic estimates of the local precision level required.

In our opinion, two areas deserve further investigation: (a) the propagation of inherent errors in highly nonlinear problems, in which a significant exchange of "error energy" among spectral components can take place, and (b) the establishment of statistical properties of sequential local error occurrences in large-scale, ill-conditioned systems. The study of either subject is likely to require extensive numerical experimentation.

Section 11
REFERENCES

1. Henrici, P., Discrete Variable Methods in Ordinary Differential Equations, Wiley, New York, 1962.

2. Henrici, P., Error Propagation for Difference Methods, Wiley, New York, 1963.

3. Jensen, P. S., "Transient Analysis of Structures by Stiffly Stable Methods," J. Comp. Struct. 4, 1974, pp. 615-626.

4. Gear, C. W., Numerical Initial Value Problems in Ordinary Differential Equations, Prentice-Hall, Englewood Cliffs, N. J., 1971.

5. Dalquist, G. and Björk, A., Numerical Methods, Prentice-Hall, Englewood Cliffs, N. J., 1975.

6. Newmark, N. M., "A Method of Computation for Structural Dynamics," J. Eng. Mech. Div. ASCE, 85, 1959, pp. 67-94.

7. Houbolt, J. C., "A Recurrence Matrix Solution for the Dynamic Response of an Elastic Aircraft," J. Aero Sci. 17, 1950, pp. 540-550.

8. Wilson, E. L., Farhoomand, I., and Bathe, K. J., "Nonlinear Dynamic Analysis of Complex Structures," Earthquake Eng. and Struct. Dyn., 1, 1973, pp. 241-252.

9. Park, K. C., "An Improved Stiffly Stable Method for Direct Integration of Nonlinear Structural Dynamics," J. Appl. Mech., 42, 1975, pp. 464-470.

10. Park, K. C. and Geers, T. L., "A Matrix Scaling Technique for Efficient Transient Structural Response Analysis," to be presented at the Int. Symp. on Innovative Numerical Analysis in Applied Engineering Sciences, Versailles (France), May 23-27, 1977.

11. Geradin, M., "A Classification and Discussion of Integration Operators for Transient Structural Response," AIAA Aerospace Sciences Meeting, Washington, D. C., January 1974.

12. Wilkinson, J. H., Rounding Errors in Algebraic Processes, Prentice-Hall, Englewood Cliffs, N. J., 1963.

13. Kalman, R. E., "A New Approach to Linear Filtering and Prediction Problems," J. of Basic Eng., Trans. ASME, March 1960, pp. 35-45.

14. Hughes, T. J. R., "Stability, Convergence, and Growth and Decay of Energy of the Average Acceleration Method in Nonlinear Structural Dynamics," J. Comp. Struct., 6, 1976, pp. 313-322.

15. Jensen, P. S., "Stiffly Stable Methods for Undamped Second-Order Equations of Motion," SIAM J. Numer. Anal., 13, 1976, pp. 549-563.

16. Goudreau, G. L., and Taylor, R. L., "Evaluation of Numerical Integration Methods in Elastodynamics," Computer Meth. in Appl. Mech. and Engng., 2, 1972, pp. 69-97.

17. Nayfeh, A. H., Perturbation Methods, Wiley, New York, 1973.

18. Chan, S. P., Cox, H. L., and Benfield, W. A., "Transient Analysis of Forced Vibration of Complex Structural Mechanical Systems," J. Royal Aero. Soc., 6, 1972, pp. 457-460.

ACKNOWLEDGMENTS

The authors wish to express their appreciation to Dr. T. L. Geers, who suggested the investigation of computational aspects of integration procedures, and to Dr. J. A. DeRuntz and Philip Underwood for many helpful suggestions. The preparation of this paper has been supported by the Independent Research program of Lockheed Missiles and Space Company, Inc., and by the Office of Naval Research under Contract N00014-74-C-0355.

Appendix C

NUMERIC QUANTITIES FOR SPECIFIC OPERATORS

(1) Integrator Coefficients for Some One-Derivative Methods, Eq. (7a)

Operator (Stability*)	m	Coefficients α_i, β_i, i = 0, ... m				
		α/β	i = 0	i = 1	i = 2	i = 3
Backward Euler (A)	1	α	1	-1		
		β	1	0		
Gear 2-Step (A)	2	α	1	-4/3	1/3	
		β	2/3	0	0	
Gear 3-Step (S)	3	α	1	-18/11	9/11	-2/11
		β	6/11	0	0	0
Jensen F3**(A)	3	α	1	-1.92601	1.13841	-0.21240
		β	0.52503	-0.02916	-0.29085	0.08136
Park 2-Step***(A)	2	α	1	-1.2	0.2	
		β	0.6	0	0	
Park 3-Step (A)	3	α	1	-1.5	0.6	-0.1
		β	0.6	0	0	0
Trapezoidal Rule (WA)	1	α	1	-1		
		β	0.5	0.5		

* (A) = A-stable, (WA) = Weakly A-stable, (S) - Stiffly stable.

** Formula 3 in [15]; includes recommended numerical damping to attain A-stability.

*** Two-step member of starting family (27).

(2) Integrator Coefficients for Some Two-Derivative Methods, Eq. (Alb)

Operator (Stability**)	m	Coefficients γ_i, δ_i, i = 0, ... m*				
		γ/δ	i = 0	i = 1	i = 2	i = 3
Average Acceleration (WA)	2	γ	1	-2	1	
(≡Newmark β = 1/4)		δ	1/4	1/2	1/4	
Central Difference (E)	2	γ	1	-2	1	
		δ	0	1	0	
Houbolt (A)	3	γ	1	-2.5	2	-0.5
		δ	0.5	0	0	0
Linear Acceleration (C)	2	γ	1	-2	1	
(≡Newmark β = 1/6)***		δ	1/6	4/6	1/6	
Wilson, θ = 1.37 (A)	3	γ	1	-2.27007	1.54015	-0.27007
		δ	0.31282	0.36820	0.05507	-0.00616
Wilson, θ = 2 (A)	3	γ	1	-2.5	2	-0.5
		δ	2/3	-5/12	1/3	-1/12

* $\delta_i = \eta \beta_i$ in the notation of Appendix A.

** (A) = A-stable, (WA) = Weakly A-stable, (C) = Conditionally stable, (E) = Explicit

*** Also coincides with Wilson, θ = 1.

(3) Coefficients ρ and σ Occurring in Asymptotic Expansions (50) and (62), for Specific Operators and Computational Paths

Operator	ρ/σ*	Path			
		(0)	(0')	(1)	(2)
Backward Euler	ρ	0	0	1.0	1.0
	σ	1.0	1.0		
Gear 2-Step	ρ	0	0	2.25	2.25
	σ	1.5	1.5		
Gear 3-Step	ρ	0	0	3.36	3.36
	σ	1.89	1.89		
Jensen F3	ρ	0	0	6.65	3.63
	σ	1.905	3.47		
Park 2-Step	ρ	0	0	2.08	2.78
	σ	1.67	1.25		
Park 3-Step	ρ	0	0	2.78	2.78
	σ	1.67	1.67		
Trapezoidal Rule	ρ	0	0	2.0	4.0
	σ	2.0	1.0		
Average Acceleration	ρ	1.0			4.0
Central Difference**	ρ	1.0			
Houbolt	ρ	2.0			2.0
Linear Acceleration	ρ	1.0			6.0
Wilson, θ = 1.37	ρ	1.37			3.20
Wilson, θ = 2	ρ	2.0			1.50

* The value of σ is given only if ρ = 0, i.e., expansion (62) holds.

** Computational path distinction does not apply to explicit methods.

Appendix D
POST SCRIPTUM: THE WAY IT WAS

The organization of the main body of this paper follows the traditional style of scientific writing, viz., the presentation of a chiefly analytical body of theory, followed by its experimental substantiation. Experienced researchers know, however, that the processes of experimentation and theory development usually interweave in a complex manner. The formal presentation of computationally oriented papers is, in many respects, a holdover from the pre-computer era. Numerical experiments weɪe then performed either to demonstrate the validity of a proposed theory or merely to tabulate derived results. Computational mechanics has by now evolved to a stage in which exploratory computations cannot only offer direct insight into physical processes (numerical simulation), but may lead the way into the establishment of mathematical theories. In this regard, computational work is gradually complementing, if not actually replacing, conventional experimental methods. Note that we have labeled Section 8 of Part II "Numerical Experiments" rather than "Numerical Results." In this Appendix, we shall attempt to describe, in rough chronological order, what we were trying to find out, what we did, and why, and when. We hope that the readers (and the writers will become such after this is written!) will identify with and benefit from this candid reconstruction of the course (and curses) of a research project.

The Setting

As stated in the Introduction to Part II, our main objective was to find a computer implementation form, within the class of one-derivative implicit integration procedures, which requires minimal computational effort per step and yet achieves satisfactory error propagation control. Three implementations were initially considered: the (f - Ku) form, the (Mü) form, and the conventional implementation of the Newmark family; these were later classified as (J0), (C1), and (C2) forms, respectively, in the nomenclature of Part I.

Another departure point was an awareness of the existence of techniques for assessing the propagation of rounding errors in the conventional numerical solution of first-order initial value problems. Virtually all of the existing theory has been developed by Henrici [1, 2]. A strenuous journey through Reference [1] revealed, however, that the class of computational errors we had in mind was not dealt with at all. The error sources we were mostly concerned about included starting errors, uncertainties in the evaluation of mass and stiffness matrices, and algebraic solution errors arising from inaccurate matrix factorizations (in linear problems) or extrapolation/iterative schemes (in nonlinear problems), in addition to the rounding errors. Moreover, Henrici did not stress the distinction between propagated and accumulated error, neither did mention the frequency-decomposition approach on which most of our work was to be based.

With these "initial supplies" behind us, we plunged into the research labyrinth, through which we progressed by a combination of iterative experimentation, theory development, and some degree of luck.

The Initial Adventure

As a starting point, the one-step method family

$$u_n = u_{n-1} + h\,[f\,\dot{u}_n + (1 - f)\,\dot{u}_{n-1}] \qquad 0 < f \leq 1 \tag{D1}$$

was used to treat a single, undamped linear oscillator

$$\ddot{u}_n + \omega^2 u_n = 0 \tag{D2}$$

adopted as model error propagation equation (u here denotes displacement error). The initial conditions were:

$$\begin{aligned} u_o &= 1, & u_{-1} &= \dots = u_{-\infty} = 0 \\ \dot{u}_o &= 1/(hf), & u_{-1} &= \dots = \dot{u}_{-\infty} = 0 \end{aligned} \tag{D3}$$

These conditions were justified as follows. An assumed unit error source in the displacement occurs at a typical time station, which can be taken to be t = 0 for conveniency. To study the propagation of this error source, we simply ignore error sources at other stations as well as the computed solution on account of the linearity of Eq. (D2); this scenario leads to (D3). Proceeding through the first time step of the (C1) form (see Table 1) gives

$$
\begin{aligned}
u_o &= -\omega^2 u_o = -\omega^2 \\
h_1^{\dot{u}} &= \dot{u}_o + (1-f)\, h\, \ddot{u}_o = [1 - f(1-f)\, \omega^2 h^2]/(h\, f) \\
h_1^{u} &= u_o + (1-f)\, h\, \dot{u}_o = 1/f \\
g_1 &= 1/f + [1 - f(1-f)\, \omega^2\, h^2] && \text{(D4)} \\
u_1 &= g_1/(1 + f^2\, \omega^2\, h^2) \rightarrow 1 + 1/f \quad \text{as } \omega h \rightarrow 0 \\
u_1 &= (u_1 - h_1^{u})/(h\, f) \rightarrow 1/f \quad \text{as } \omega h \rightarrow 0
\end{aligned}
$$

Another cycle produces the values

$$
\left.\begin{aligned}
u_2 &\sim 1 + 2/f \\
\dot{u}_2 &\sim 1/f
\end{aligned}\right\} \quad \text{as } \omega h \rightarrow 0 \qquad \text{(D5)}
$$

The preceding calculations suggested that the displacement error source (D3a) is amplified on subsequent stations if the (C1) form is used. The next obvious question is: how does the peak amplitude depend on ωh? This question was posed to the computer. In this instance, numerical experiments were preferred over analytical procedures on account of expediency (quick answers are very important in the initial stages of a research project), and the likelihood of crippling mistakes being committed in long hand calculations. The computer output indicated that peak amplitudes were proportional to $(\omega h)^{-1}$ in the small stepsize range, a result obviously in harmony with the trend suggested by (D4) and (D5).

The (J0) form was then programmed and run through the amplification analysis. The results were puzzling. Peak amplitudes for the trapezoidal rule (f = 0.5) and backward Euler method (f = 1) remained very close to two and one, respectively, even as ωh → 0. Back to pencil and paper. One cycle through the (J0) computational sequence (Table 2) gave

$$
\left.\begin{aligned}
u_1 &\sim 1/f \\
\dot{u}_1 &\sim 0
\end{aligned}\right\} \quad \text{as } \omega h \rightarrow 0 \qquad \text{(D6)}
$$

and the same asymptotic values recurred cycle after cycle. With these preliminary findings in our possession, it was time to begin establishing a more comprehensive theory.

The Reasoning

The error propagation analysis procedure typified by Eqs. (D4) through (D6) relies on the step-by-step integration of the difference equations, after which ωh is made to approach zero in order to isolate leading expansion terms. This approach is straightforward but laborious, and provides information only over as many time steps as the analyst's endurance can withstand. Moreover, specific operators have to be introduced ab initio, which is hardly conducive to the establishment of general conclusions. On the other hand, if the propagation response could be asymptotically characterized by the closed form solution of a differential equation, not only would the results be applicable over the whole time domain, but the tedious case-by-case approach would be circumvented.

We were therefore motivated to seek the conversion of the propagation difference equations to a limit differential form approached as $\omega h \to 0$. Propagation histories produced by the initial experiments suggested that $u(t_n)$ approaches a smooth harmonic behavior (cf. Figure 18). Intuitively, one would expect the difference equation (15), specialized to an undamped oscillator, to approximate the model O. D. E.

$$\ddot{u} + \omega^2 u = 0 \tag{D7}$$

whose solution satisfying the initial conditions (D3) is

$$u(t) = \cos \omega t + (\omega f h)^{-1} \sin \omega t \tag{D8}$$

As $\omega h \to 0$, the sine term is dominant and gives a peak of order $(\omega h)^{-1}$, which is the correct dependence. This result is not particularly startling, as it is well known that error difference equations have the same form as the governing difference equations. Thus, Eq. (48) was easily derived as described in Section 5 of Part II.

The error arresting properties of the (J0) form could not be explained, however, by Eq. (48). The numerical results indicated that the solution of the asymptotic error differential equation had to possess the structure

$$u(t) = u_o \cos \omega t + c \sin \omega t \tag{D9}$$

where c must be O(l) for $\omega h << 1$. Therefore, the limit form had to be insensitive to the initial velocity jump (D3b). Fortunately, we were aware of the properties of the stochastic differential equation used to model noise-free regulators. Optimal noise-free regulators possess the important property of smoothing white Gaussian noises, which are formal derivatives of Brownian motion processes. In stochastic filtering theory, such behavior is modeled by the Wiener-Hopf integro-differential equation. As the "error velocity" jump (D3b) can be viewed as the formal derivative of the initial displacement error source, it was conjectured that the limit error propagation equation of the (J0) form had to be

$$\dot{u} + \omega^2 \int_0^t u \, dt = 0 \tag{D10}$$

And so it was. Setting down all of the details of the formal derivation of Eq. (60) required many hours of brainwork, however.

Getting It All Together

After the important milestone of deriving the asymptotic error propagation equations (48) and (60) was reached, effort was directed toward the identification and categorization of local error sources. This finally resulted in Eqs. (32) and (33). An appropriate term was needed to identify displacement error sources such as (D3a), which become important in the small stepsize range. After some discussion the descriptive name "operational error" was agreed upon.

By this time, the error analysis program had grown from a small deck to one box of cards, and a set of multiple-step LMS operators had been implemented in addition to (Dl). A comprehensive test series was run using several integrators and the three basic implementations. An examination of the output revealed that the peak responses, although evincing the correct power dependence upon ωh, did not match the results of the asymptotic analysis as far as the proportionality coefficient was concerned. A study of the initial stages of the propagation history clarified this discrepancy. The initial conditions (D3) may be regarded as a unit impulse or "error shock" applied at t = 0. In computational shock dynamics, it is well known that if the spatial difference operator used to treat the numerical shock dispersion spans m grid points, then post-shock effects will be felt over 0(m) grid points. Similarly, in the case of temporal shock dispersion, it takes 0(2m) steps for the error shock to disperse when m-step one-derivative methods are used (the factor of two appears because we are dealing with second-order O.D.E. systems). After 0(2m) steps following the initial impulse, the asymptotic error differential equation is closely satisfied and a smooth harmonic response is observed. When the values (42) were used as initial conditions to be matched by the asymptotic propagation solutions (49) and (61), an excellent agreement between theory and experiments was reached. The physically suggestive term "transitional propagation period" was coined to identify the "temporal boundary layer" shock-dispersion region that bridges the impulse (D3) with the free propagation oscillatory regime. We remark that this procedure is a particular case of the "method of matched asymptotic expansions" [17].

The Beaten Path

The computer program was further expanded by the addition of two-derivative operators (which turned out to be a disappointment from the standpoint of operational error propagation control), the implementation of inherent error analysis, and the incorporation of on-line graphic display capabilities.

Eventually came the time to examine the propagation of inherent (force) errors. Analytical considerations had indicated that the propagation of this type of error sources can be significant only for large ωh, i.e., in the high frequency range. The sensitivity of the average acceleration method--as implemented by Chan, Cox, and Benfield [18]--to the propagation of high frequency error sources had been noted by other investigators. The appearance of the associated numerical beating phenomenon was therefore not surprising, and was readily explained by the elegant analytical treatment given in Section 7 of Part II. The same phenomenon was observed, however, in the trapezoidal rule implemented under the programmed (J0) form; this was unexpected and led to a careful re-examination of the implementation procedure.

The Fixing

While the authors were pondering over the trapezoidal rule beating in path (0)--supposedly the best error control implementation--a quick fix was worked out: choose $\underset{\sim}{B} = \underset{\sim}{D} + h_\beta \underset{\sim}{K}$ in (23b) instead of $\underset{\sim}{B} = \underset{\sim}{D}$. This trick was successful in that it eliminated the beating, but turned out not to be needed at all! It was discovered that the (J0) form had actually been programmed as follows:

$$
\begin{aligned}
&\cdots\cdots \\
u_n &= E^{-1} g_n \\
\dot{u}_n &= (u_n - h^u_u)/h_\beta \\
n &\leftarrow n+1 \\
\dot{v}_{n-1} &= -\omega^2 u_{n-1} \\
v_{n-1} &= h^v_{n-1} + h_\delta \dot{v}_{n-1}
\end{aligned}
\qquad \text{(D11)}
$$

At this point, v_{n-1} appears to be the equivalent of $\dot{u}_{n-1}$ for the model equation (D2), as the mass is unity. In the interest of saving a few words of memory, the program developer fell into the trap of using the same array to store $\dot{u}$ and v, which implies the (uncoded) operation

$$\dot{u}_{n-1} = v_{n-1} \qquad \text{(D12)}$$

The "past velocity correction" obviously violates the (J0) sequence listed in Table 2. The investigators of implementation procedures had fallen prey to an implementation detail!

Once the four state quantities u, $\dot{u}$, v, and $\dot{v}$ were carefully separated, the (J0) form performed as expected over the whole ωh range. We decided to label the calculation sequence involving (D12) as path (0'), which occurs naturally in the conventional (C) formulation. By then Figure 1 had been back to the drawing board for the N-th time, and we had four path offsprings instead of three, and we realized that the quick (J0) fix had actually opened the doors to another computational path family (cf. Remark in Section 4 of Part I). As in all mushrooming undertakings, there is an appropriate time to stop and appraise what has been accomplished.

The Ending

This study was initiated in earnest in August of 1976 and continued, interspersed with other activities, for the next five months. By December, Part I had been rewritten once, and Part II twice over. Several potentially interesting topics, such as the statistical accumulation of propagated error sources and the inherent error propagation in highly nonlinear problems, had been hardly touched upon. Nonetheless, 1976 was rapidly drawing to a close; it was time to prepare for the 1977 ASME meetings, the Independent Research and contract reports would be soon due, and the error analysis program hovered close to 4000 cards. So we decided to sit down and finalize the formal paper!

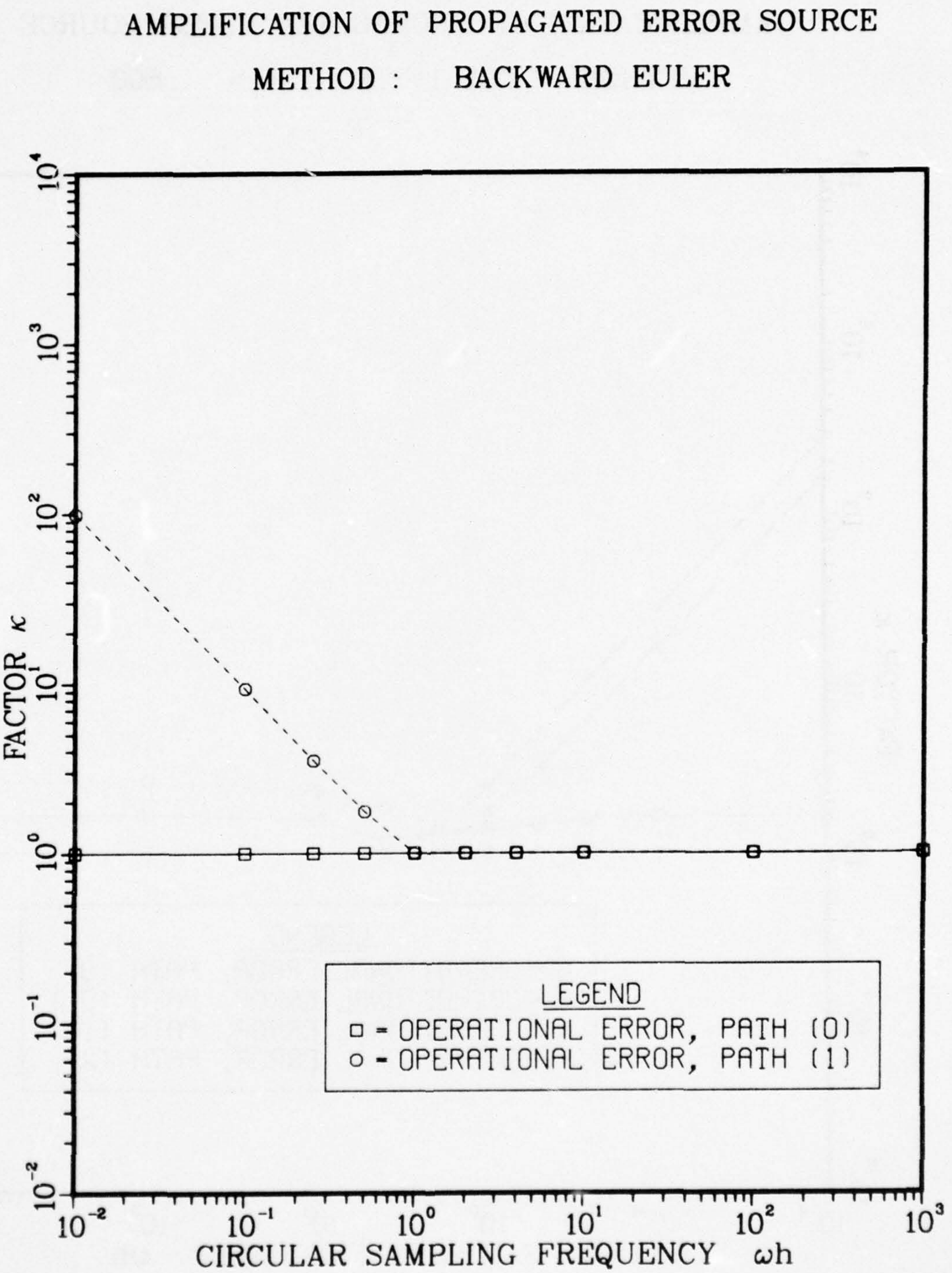


Fig. 3 Amplification spectra of propagated operational error for the backward Euler method [Paths (0') and (2) coincide with (0) and (1), respectively]

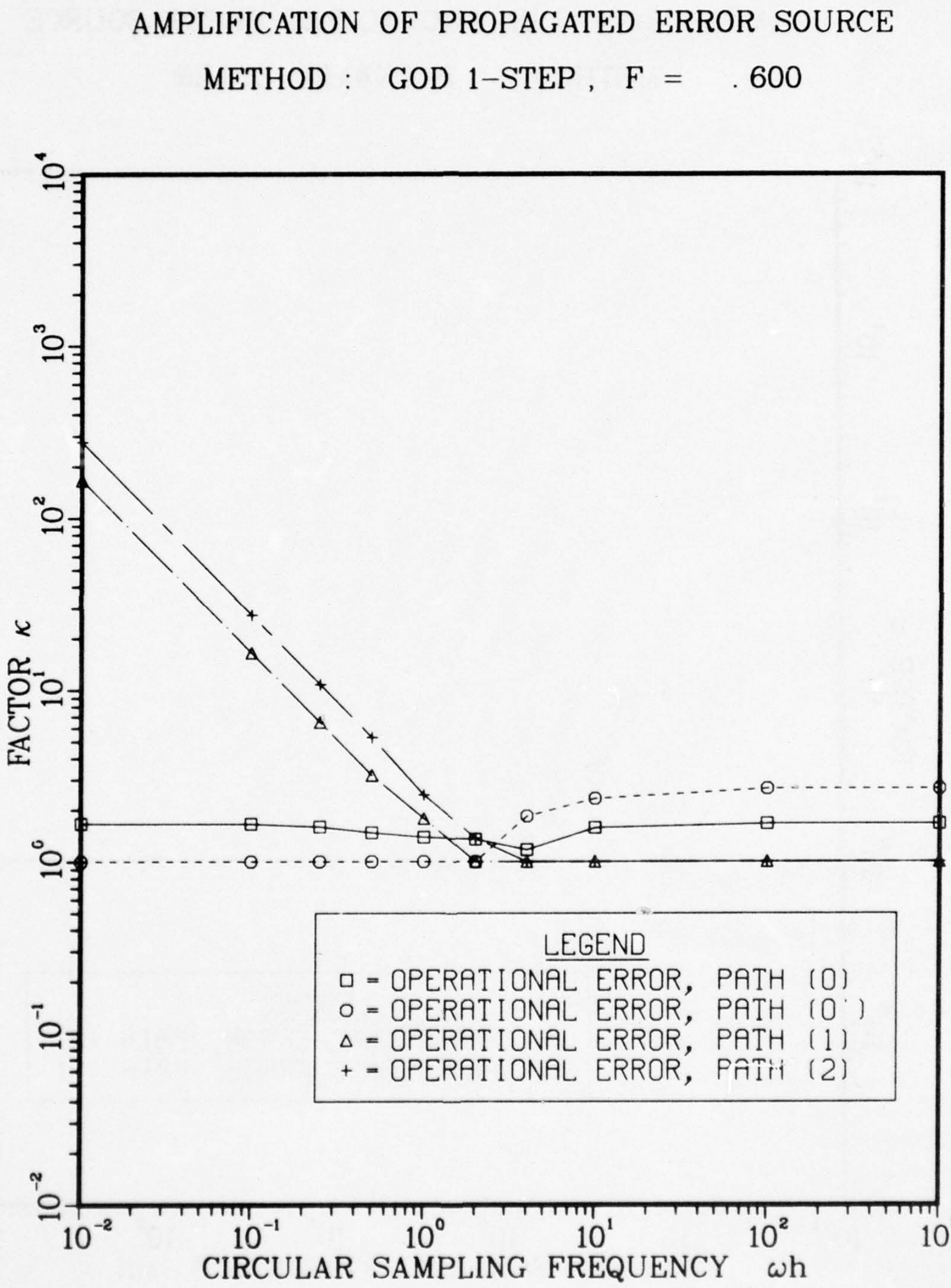


Fig. 4 Amplification spectra of propagated operational error for the general one-derivative, one-step method (79) with f = 0.60

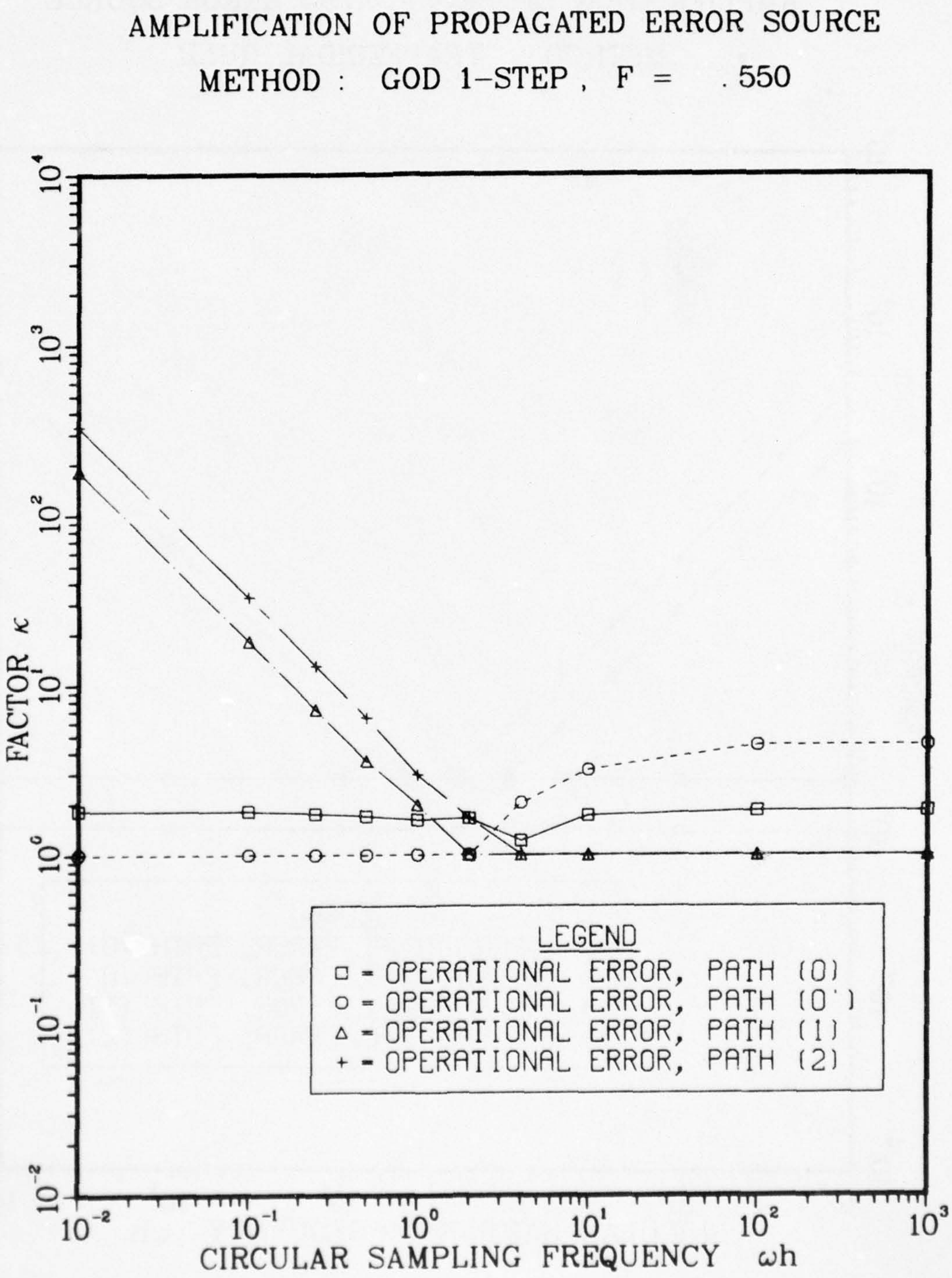


Fig. 5 Amplification spectra of propagated operational error for the general one-derivative, one-step method (79) with f = 0.55

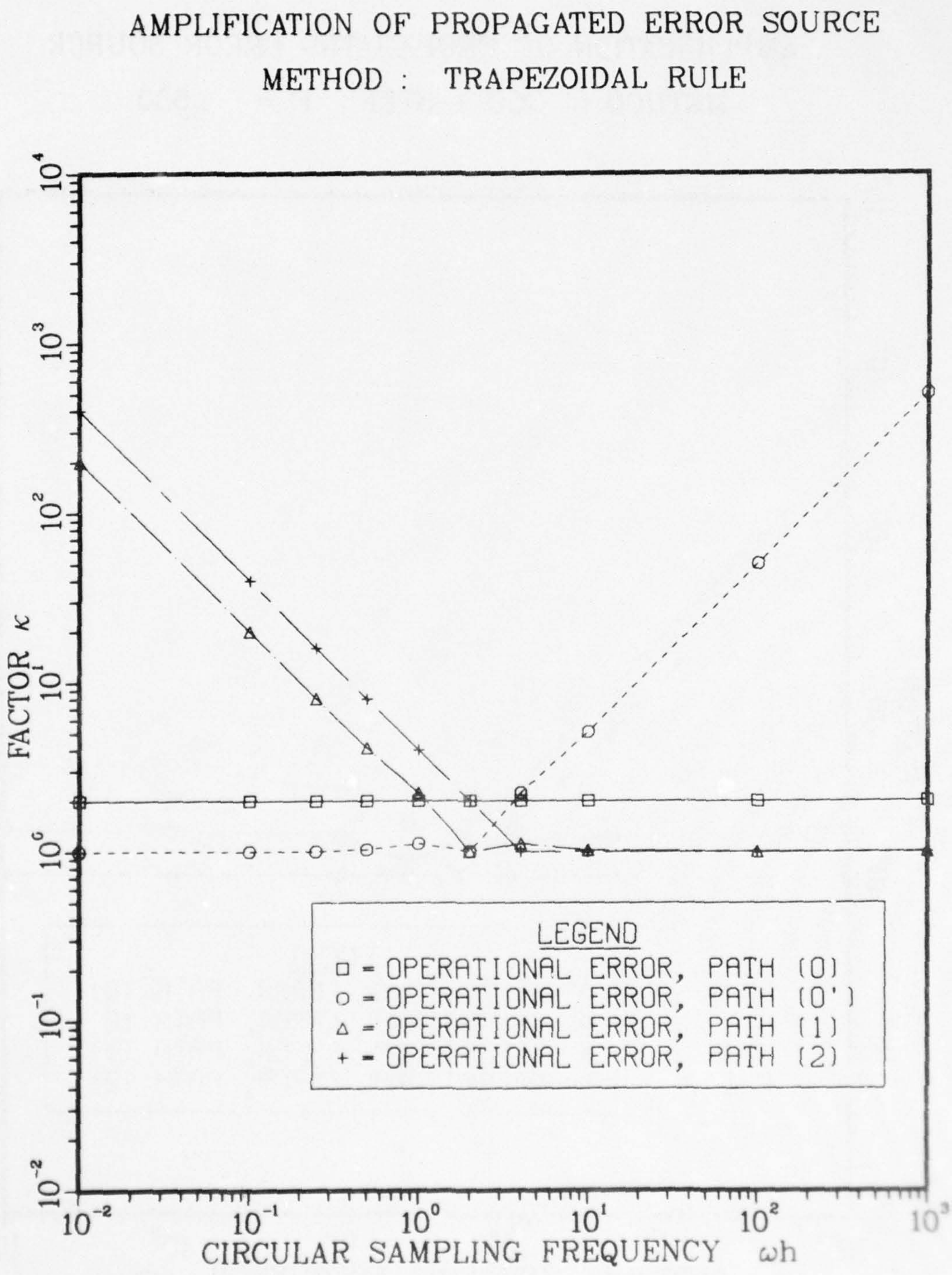
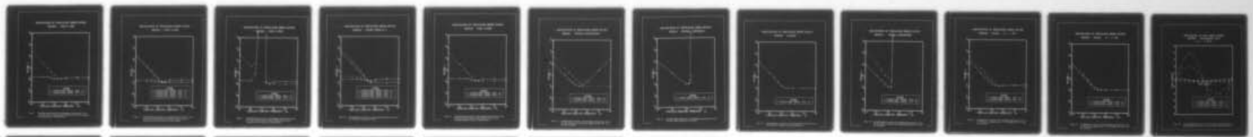
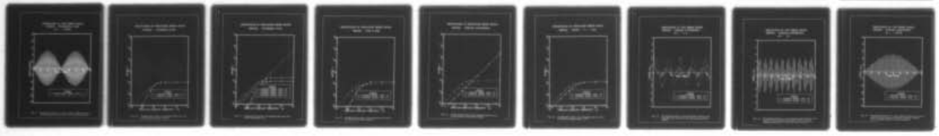


Fig. 6 Amplification spectra of propagated operational error for the trapezoidal rule

AD-A039 784 LOCKHEED MISSILES AND SPACE CO INC PALO ALTO CALIF PA--ETC F/G 9/2
COMPUTATIONAL ASPECTS OF TIME INTEGRATION PROCEDURES IN STRUCTU--ETC(U)
JAN 77 K C PARK, C A FELIPPA N00014-74-C-0355
UNCLASSIFIED LMSC/D556247 NL

2 OF 2
AD
A039784





END
DATE
FILMED
6-77

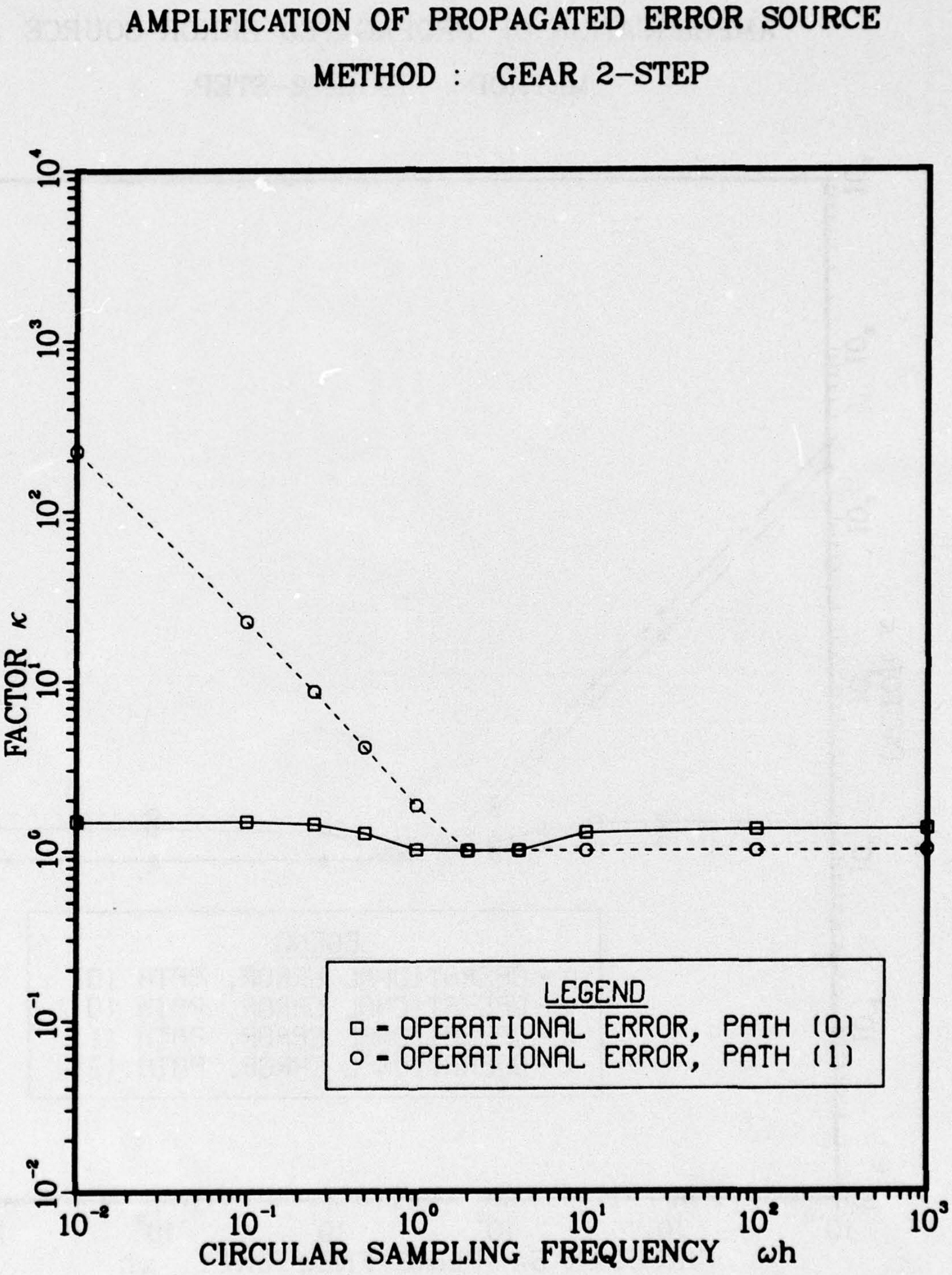


Fig. 7 Amplification spectra of propagated operational error for Gear's two-step method [Paths (0') and (2) coincide with (0) and (1), respectively]

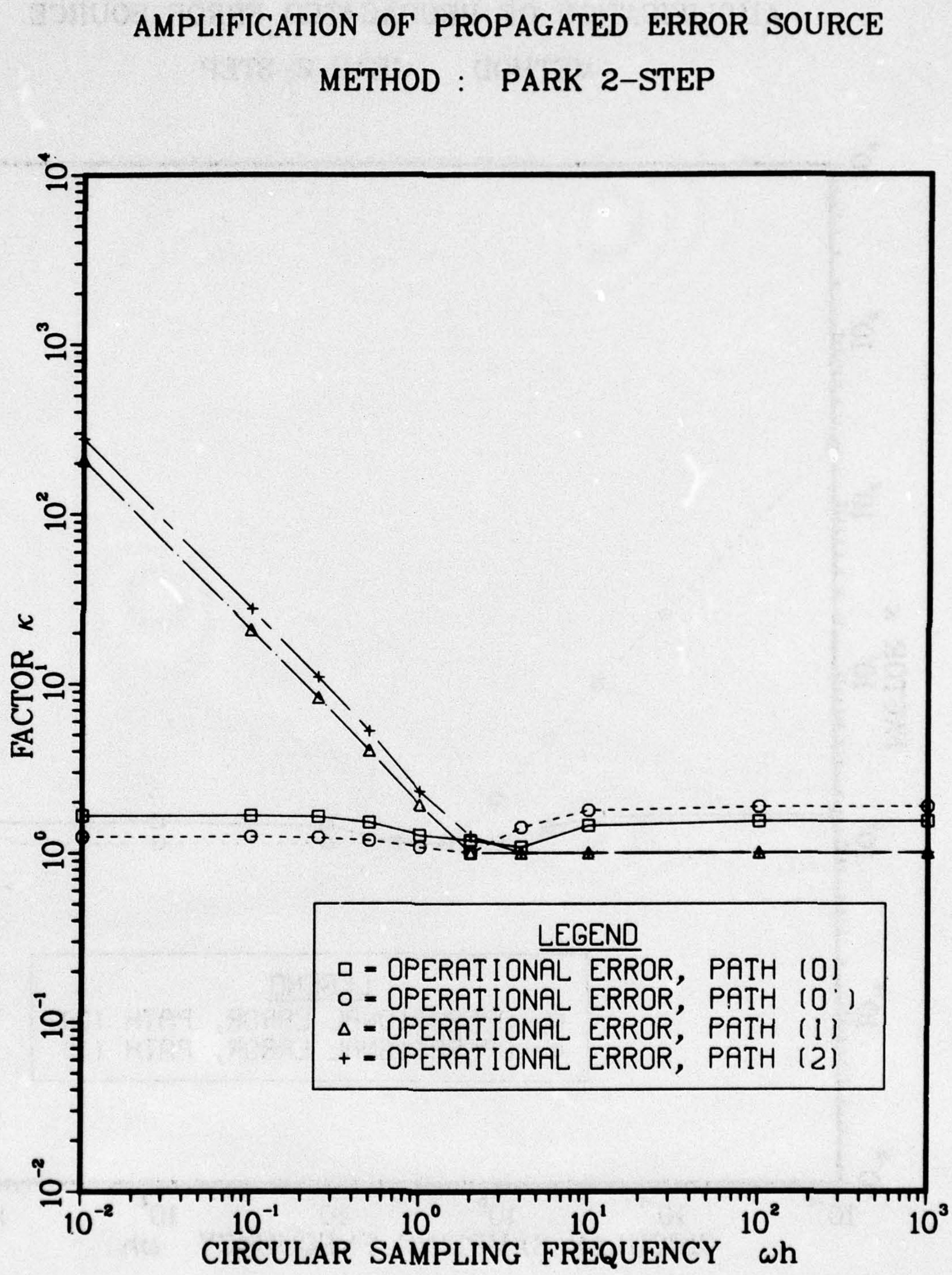


Fig. 8 Amplification spectral of propagated operational error for the two-step member of the starting operator family (27) for Park's three-step method

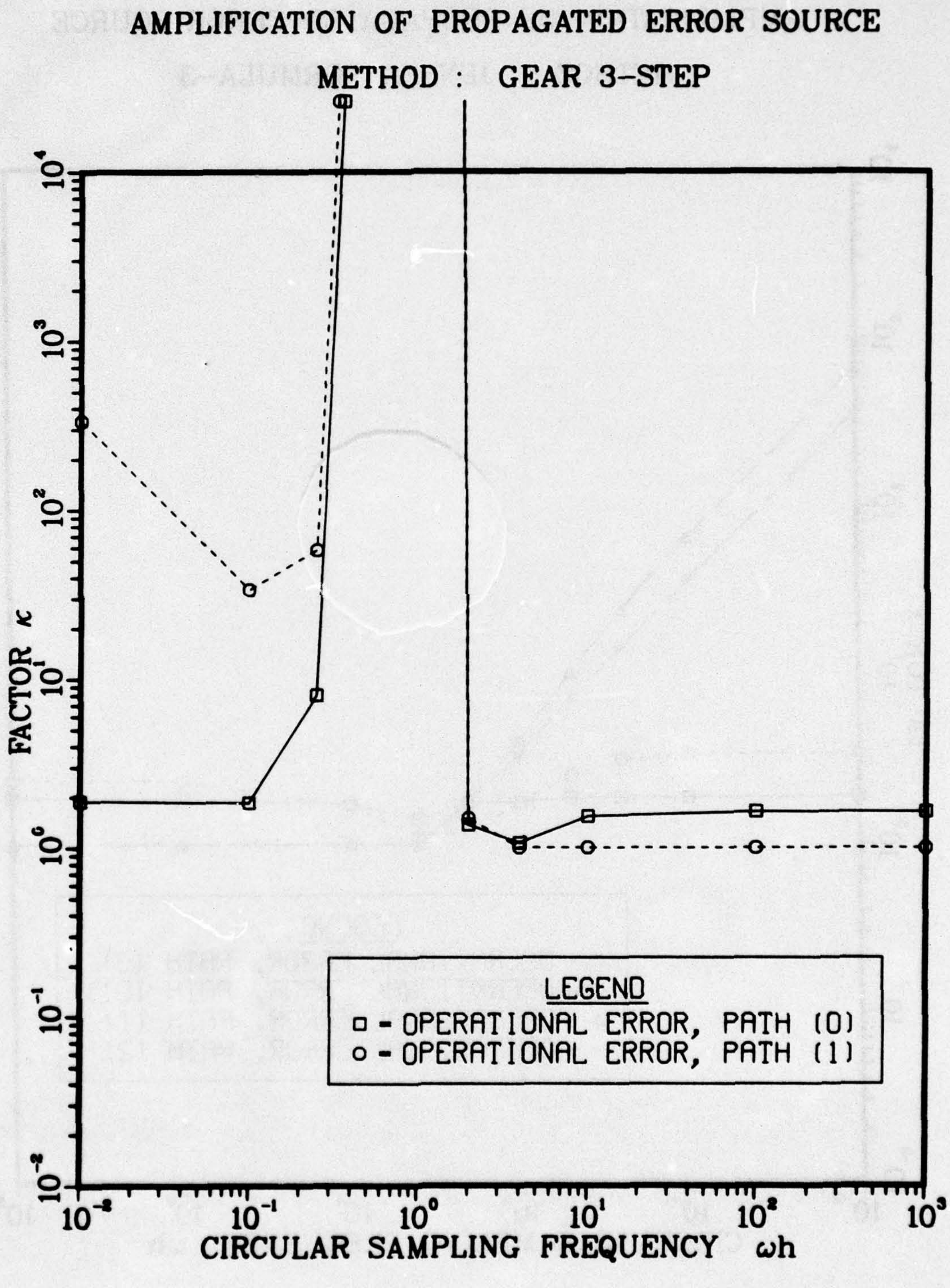


Fig. 9 Amplification spectra of propagated operational error for Gear's three-step method [Paths (0') and (2) coincide with (0) and (1), respectively]

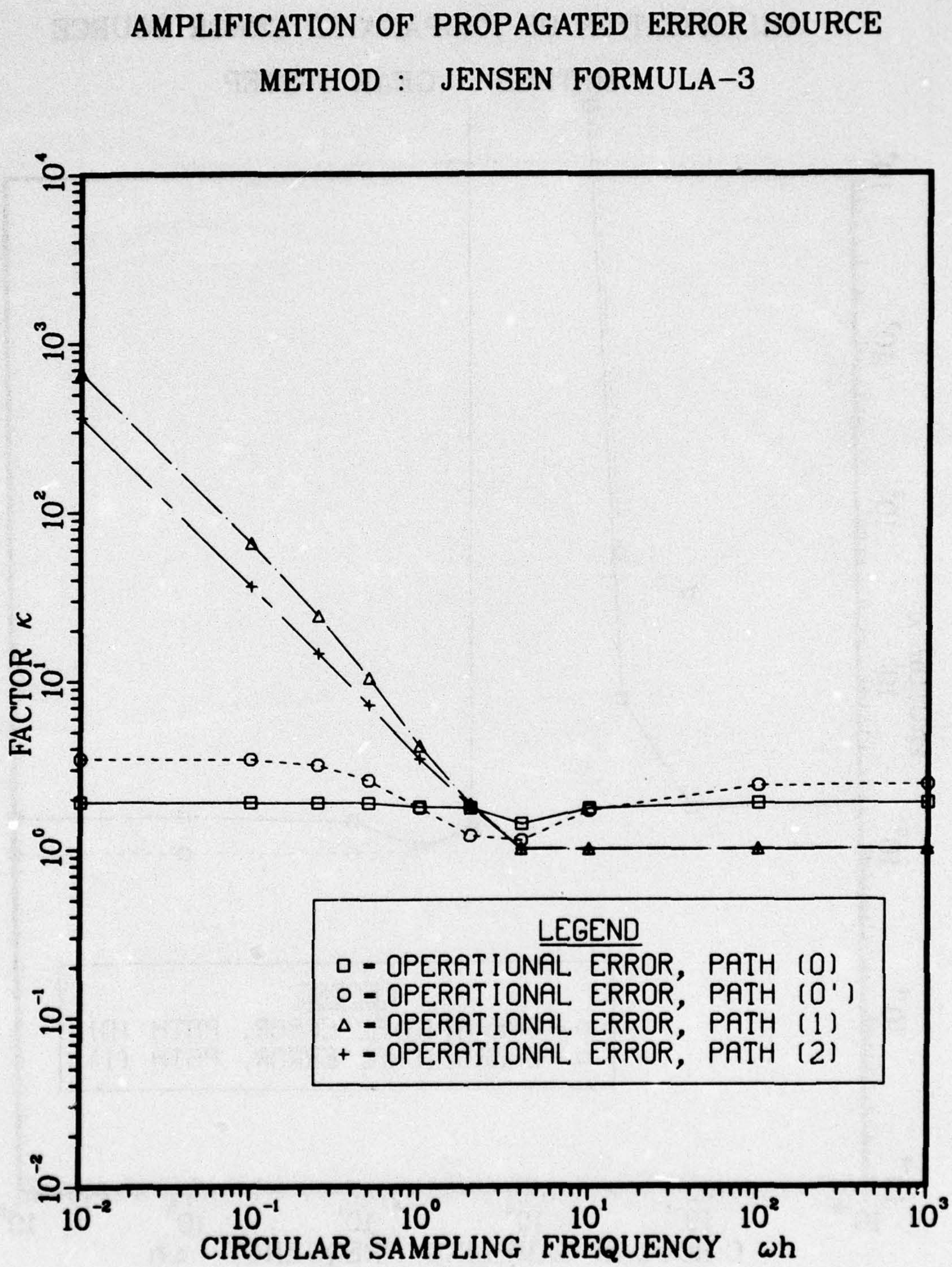


Fig. 10 Amplification spectra of propagated operational error for Jensen's Formula-3 [15]

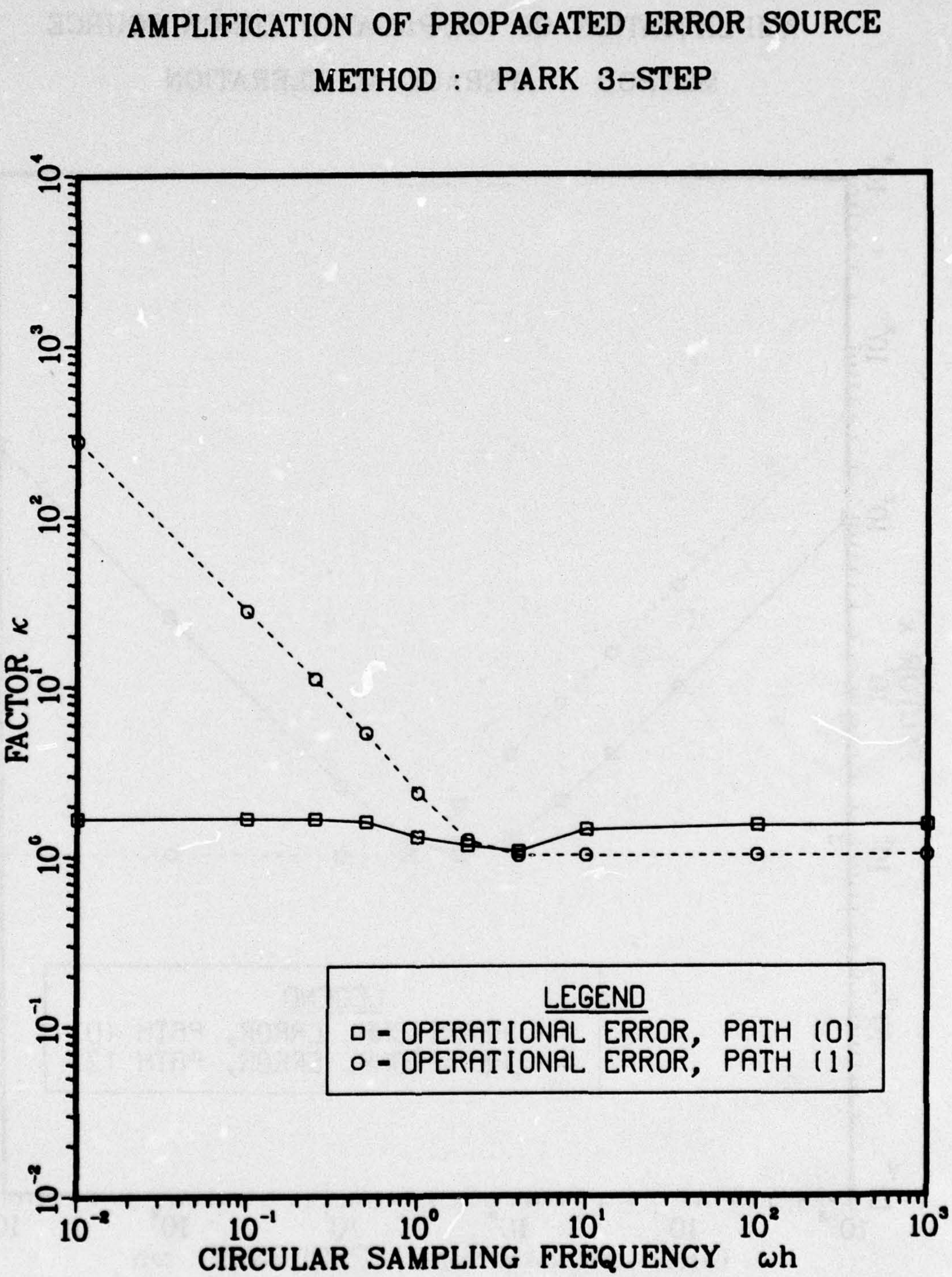


Fig. 11 Amplification spectra of propagated operational error for Park's three-step method [Paths (0') and (2) coincide with (0) and (1), respectively]

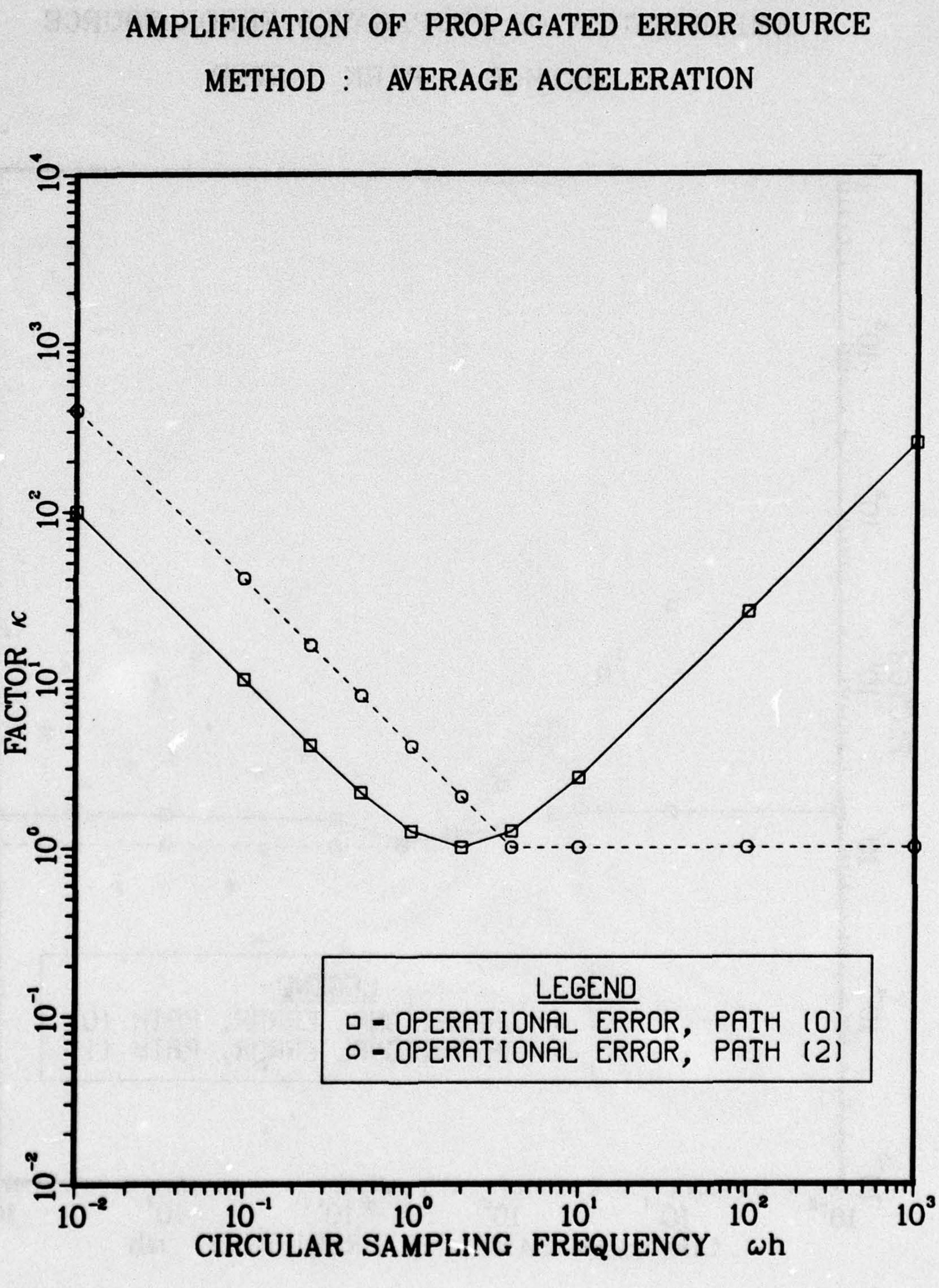


Fig. 12 Amplification spectra of propagated operational error for the average acceleration method [Paths (0), (0'), and (1) coincide]

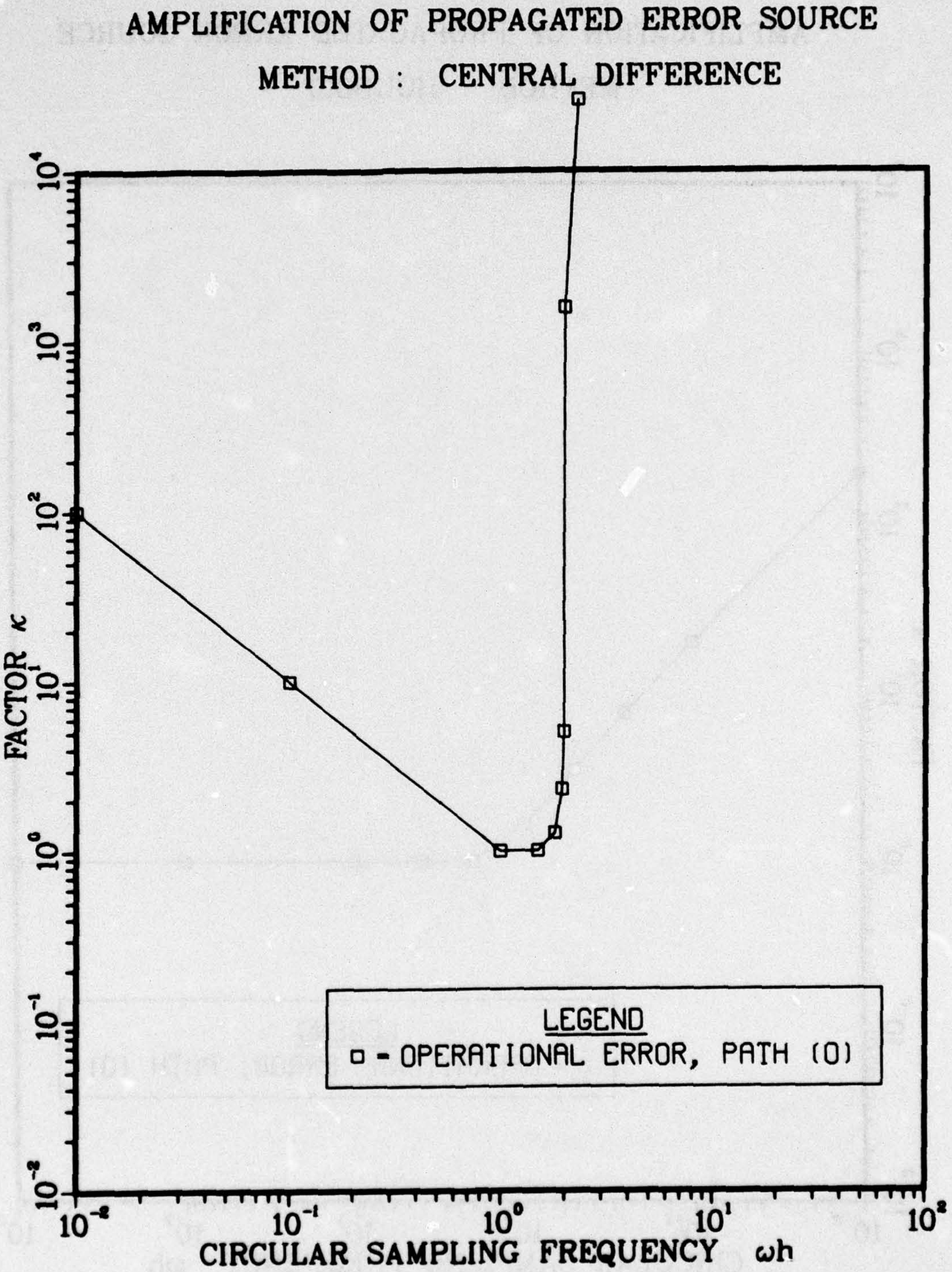


Fig. 13 Amplification spectrum of propagated operational error for the central difference method

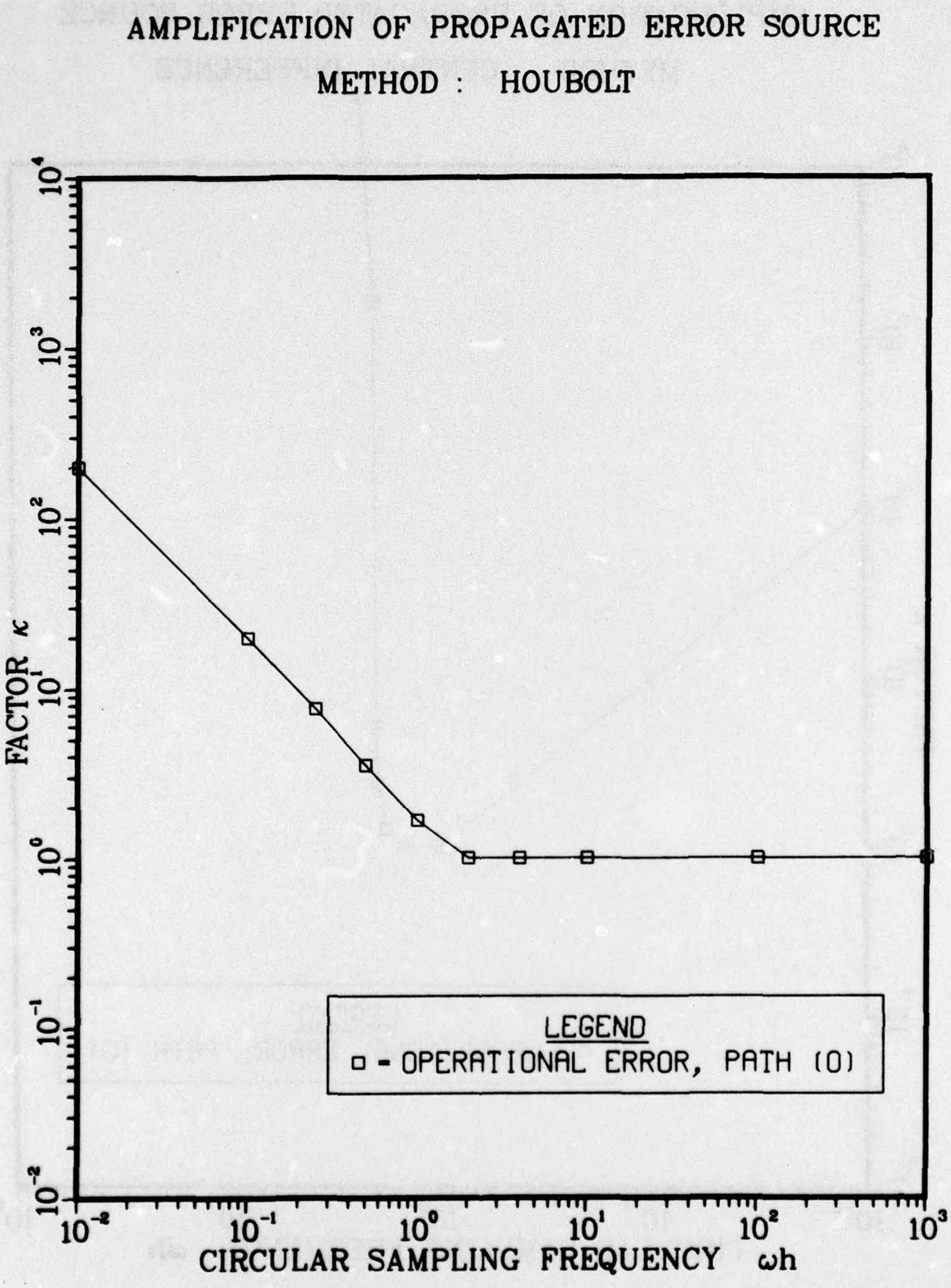


Fig. 14 Amplification spectrum of propagated operational error for Houbolt's method (all paths coincide)

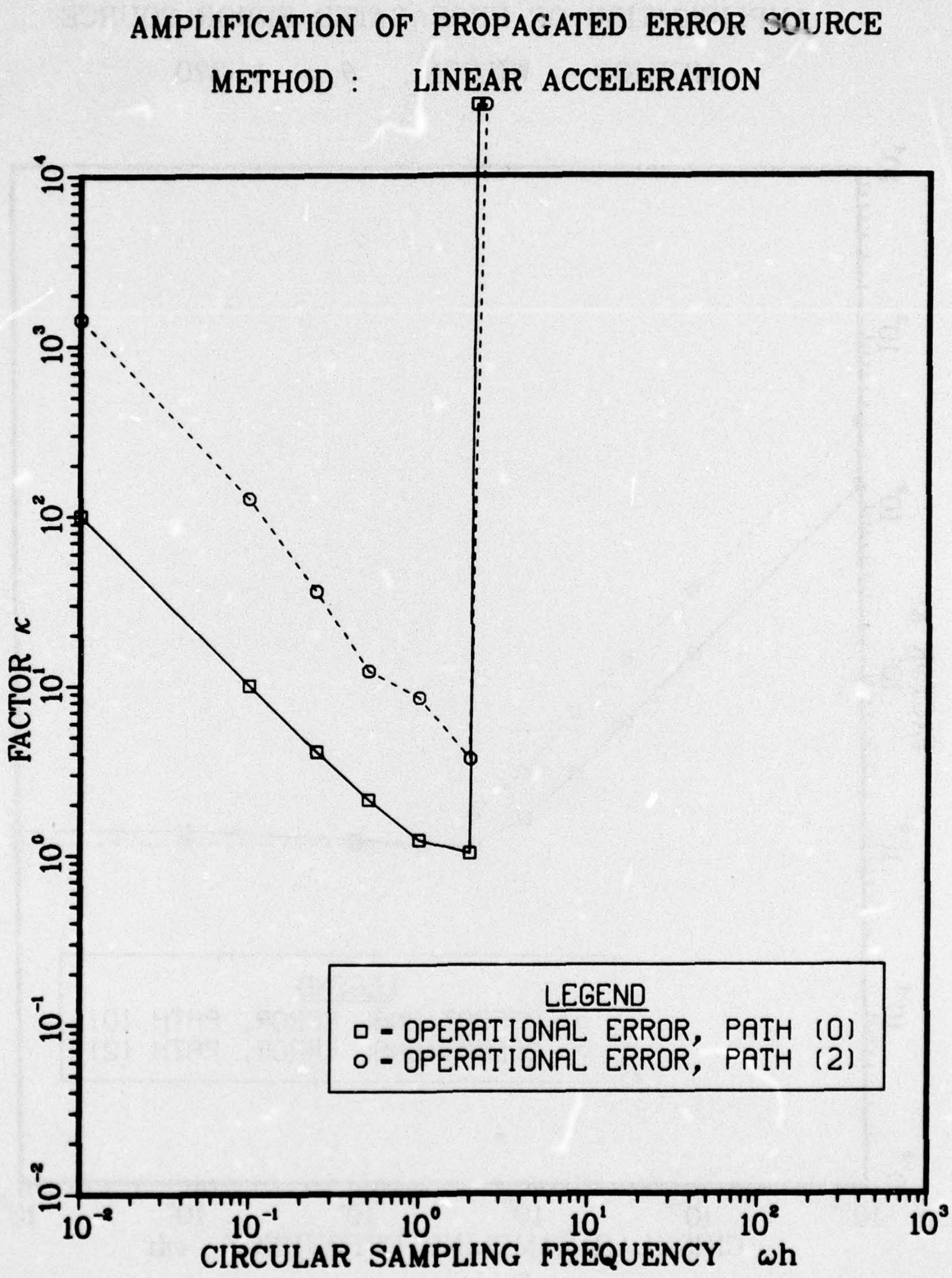


Fig. 15 Amplification spectra of propagated operational error for the linear acceleration method [Paths (0), (0') and (1) coincide]

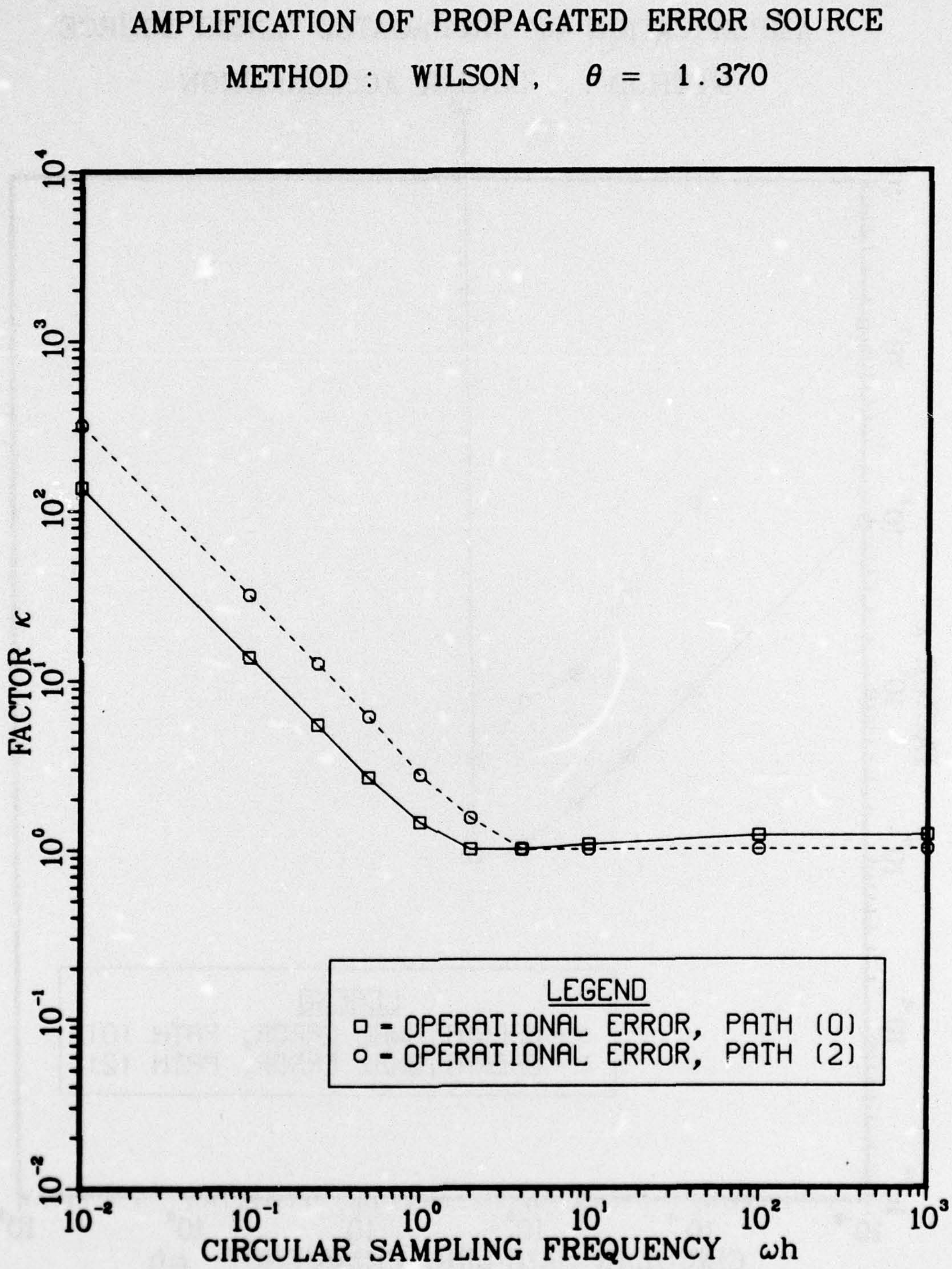


Fig. 16 Amplification spectra of propagated operational error for Wilson's method with $\theta = 1.37$ [Paths (0), (0') and (1) coincide]

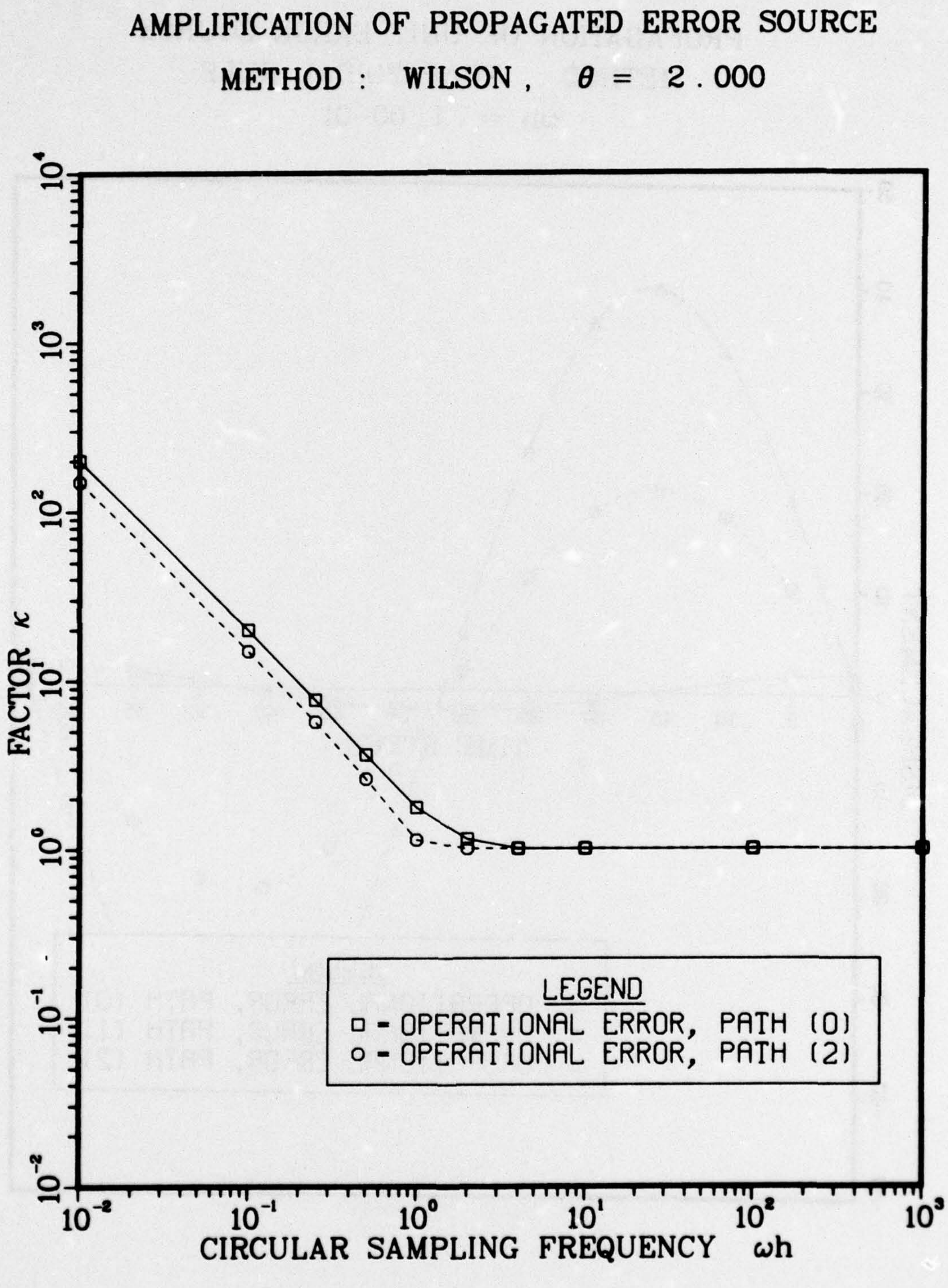


Fig. 17 Amplification spectra of propagated operational error for Wilson's method with θ = 2.0 [Paths (0), (0'), and (1) coincide]

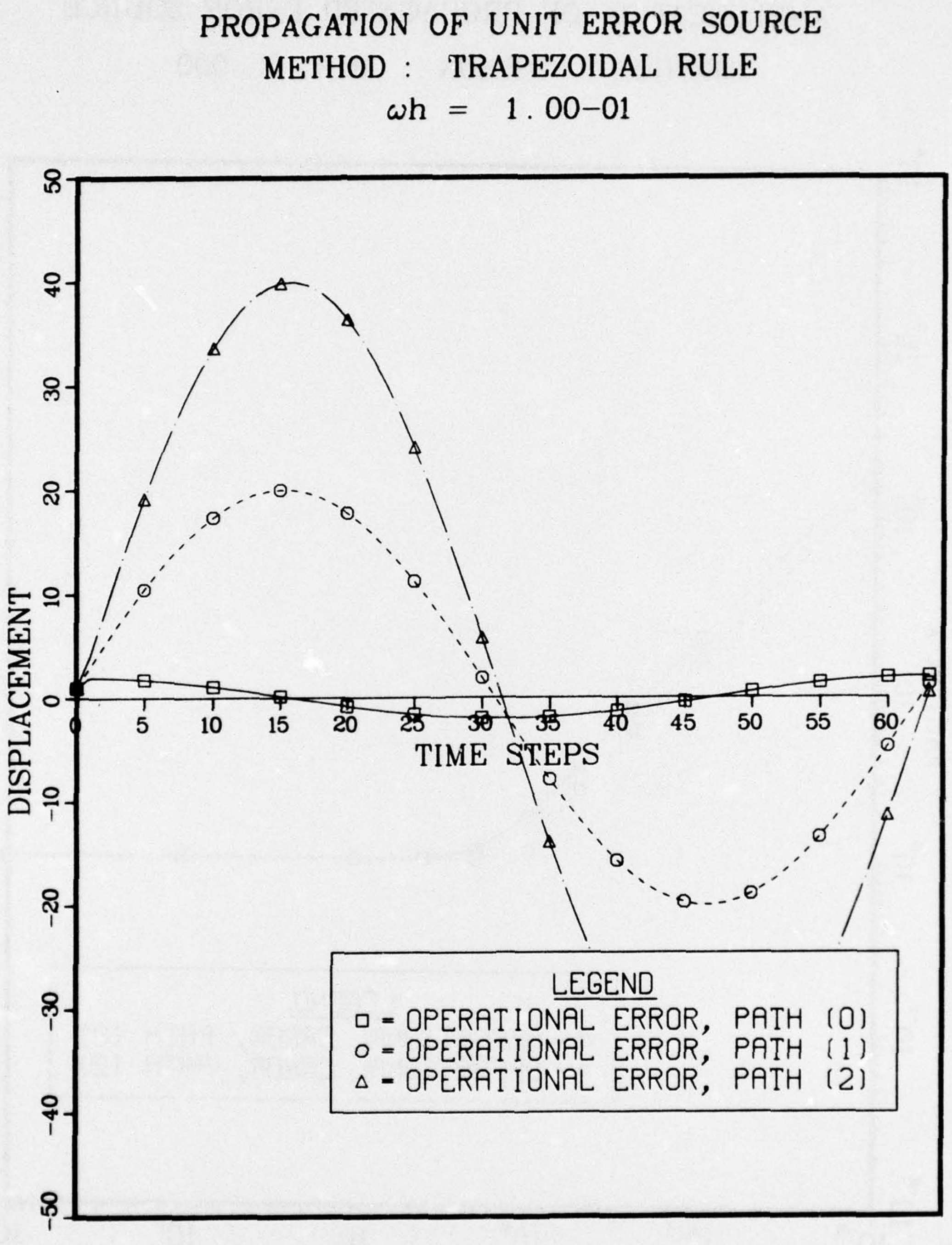


Fig. 18 Propagation history of a small stepsize (low frequency) unit operational error source for the trapezoidal rule

PROPAGATION OF UNIT ERROR SOURCE
METHOD : TRAPEZOIDAL RULE
ωh = 1.00+02

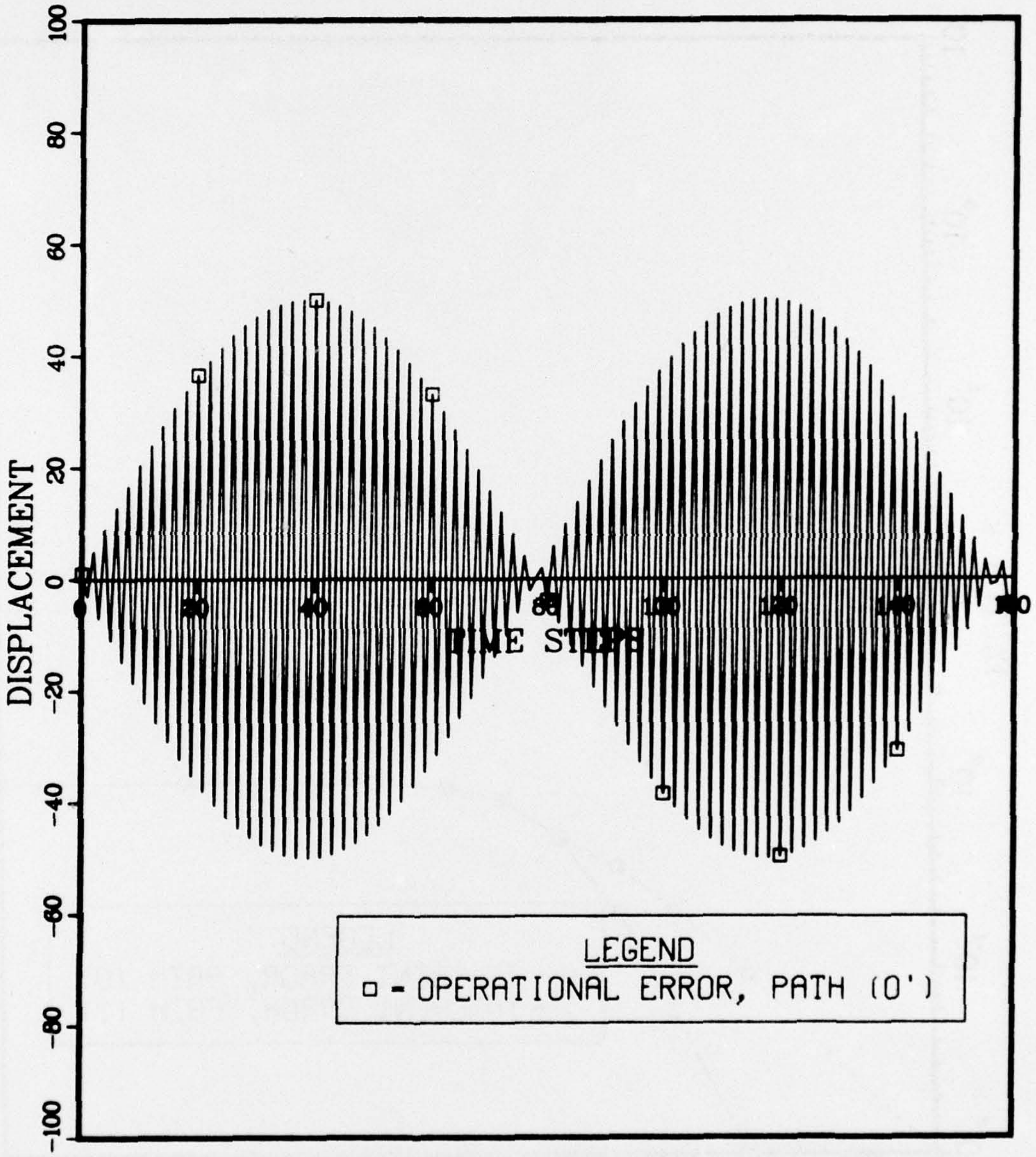


Fig. 19 Propagation history of a large stepsize (high frequency) unit operational error source for the trapezoidal rule

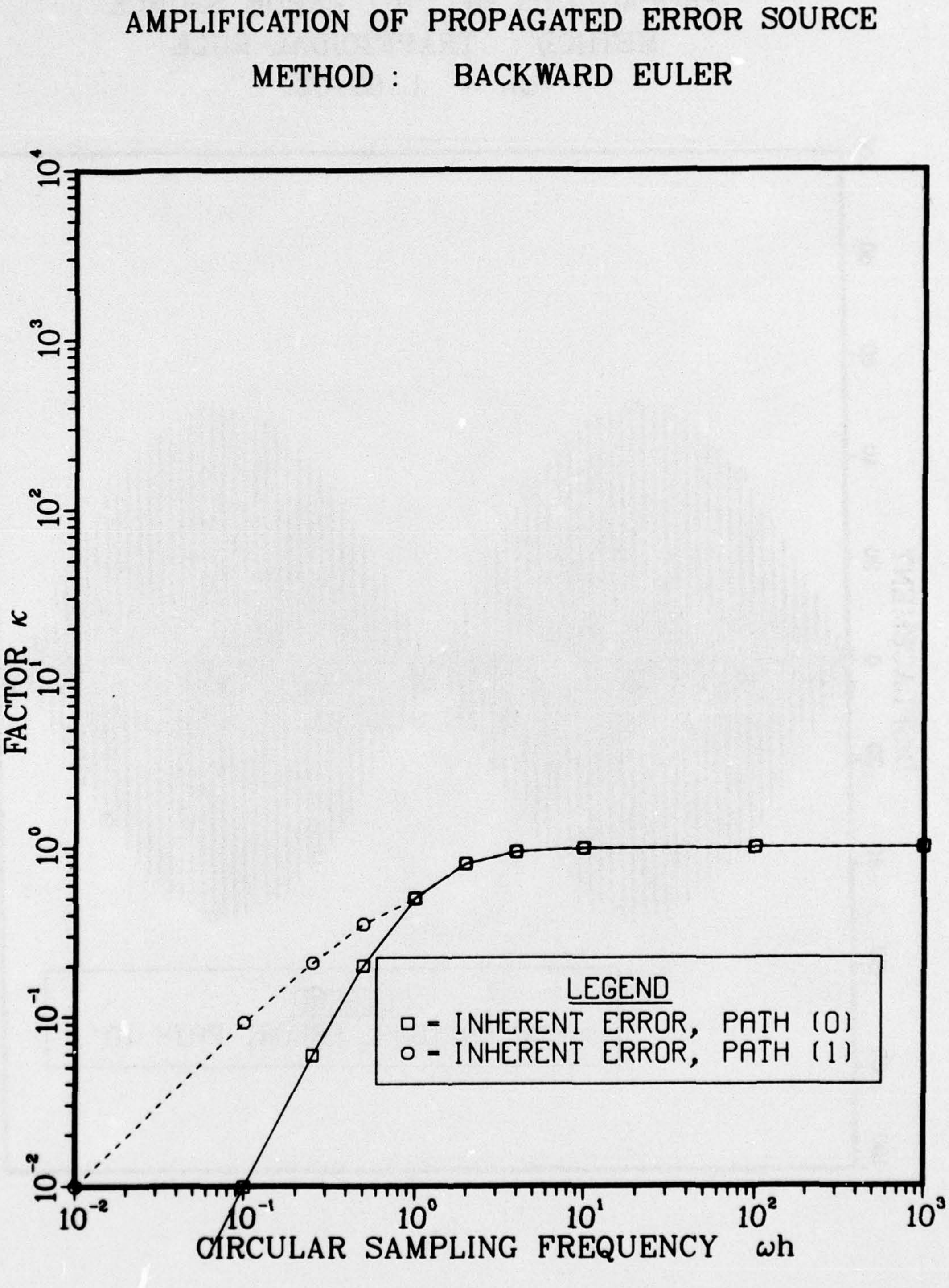


Fig. 20 Amplification spectra of propagated inherent error for the backward Euler method

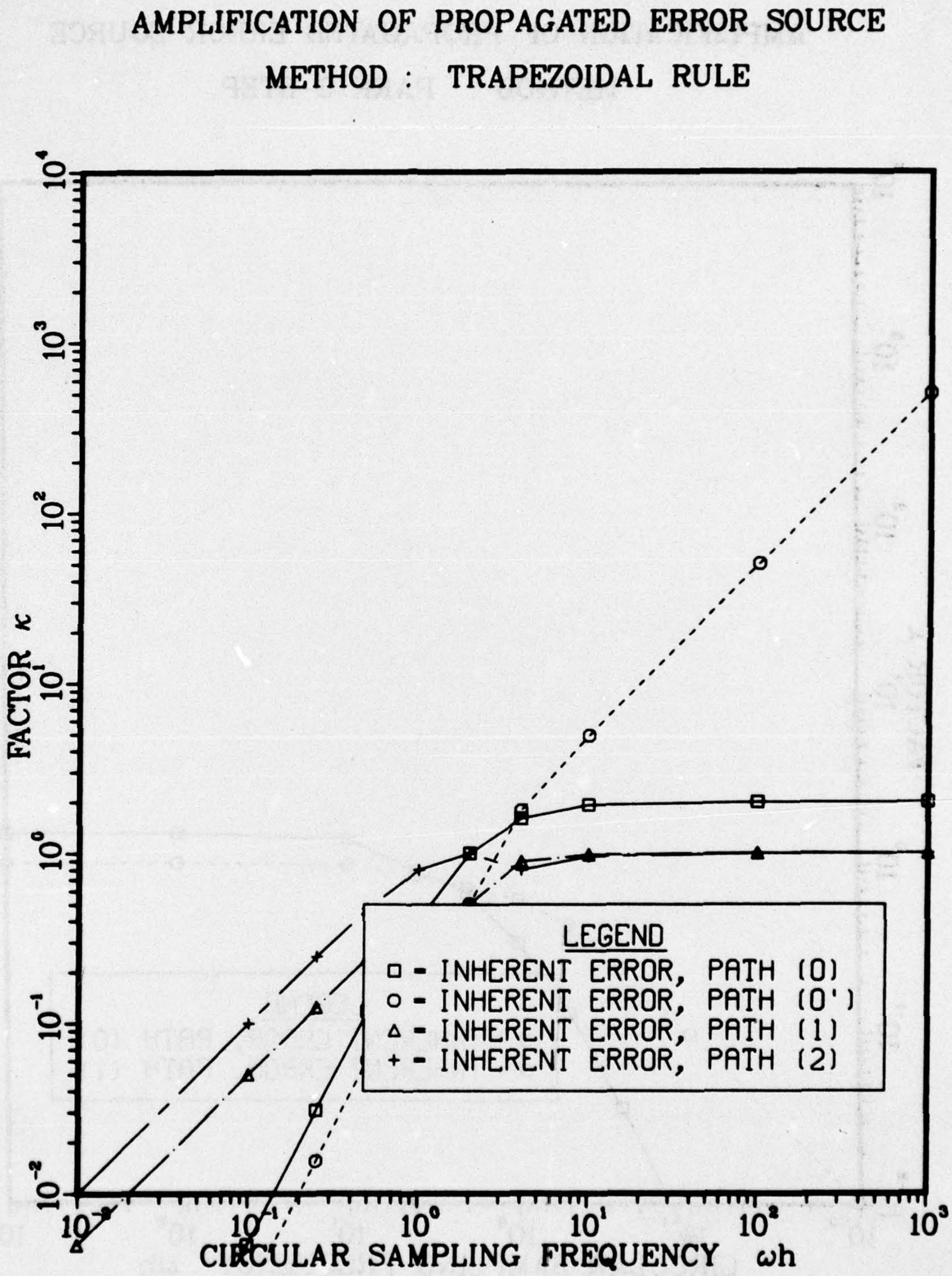


Fig. 21 Amplification spectra of propagated inherent error for the trapezoidal rule

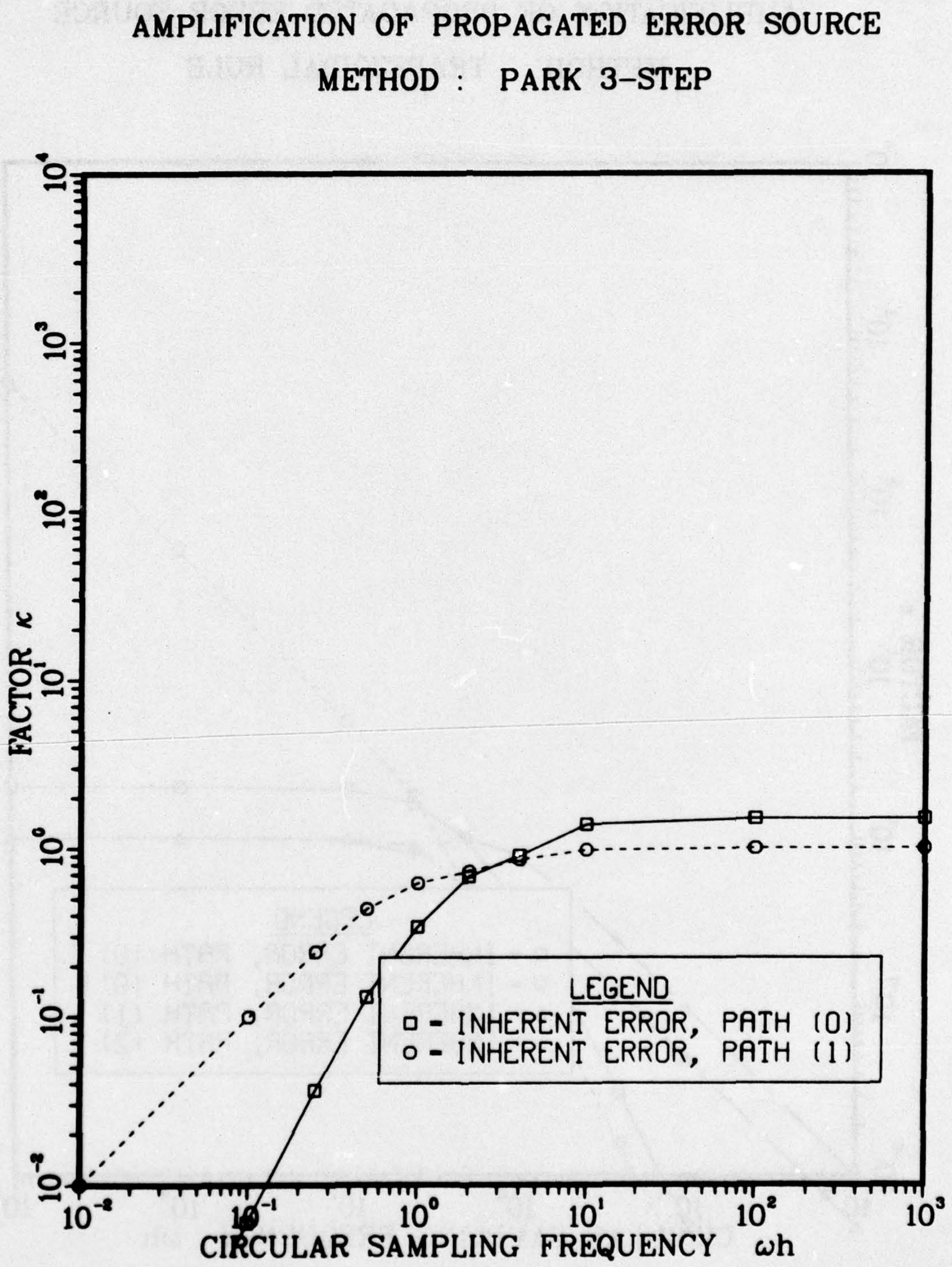


Fig. 22 Amplification spectra of propagated inherent error for Park's three-step method

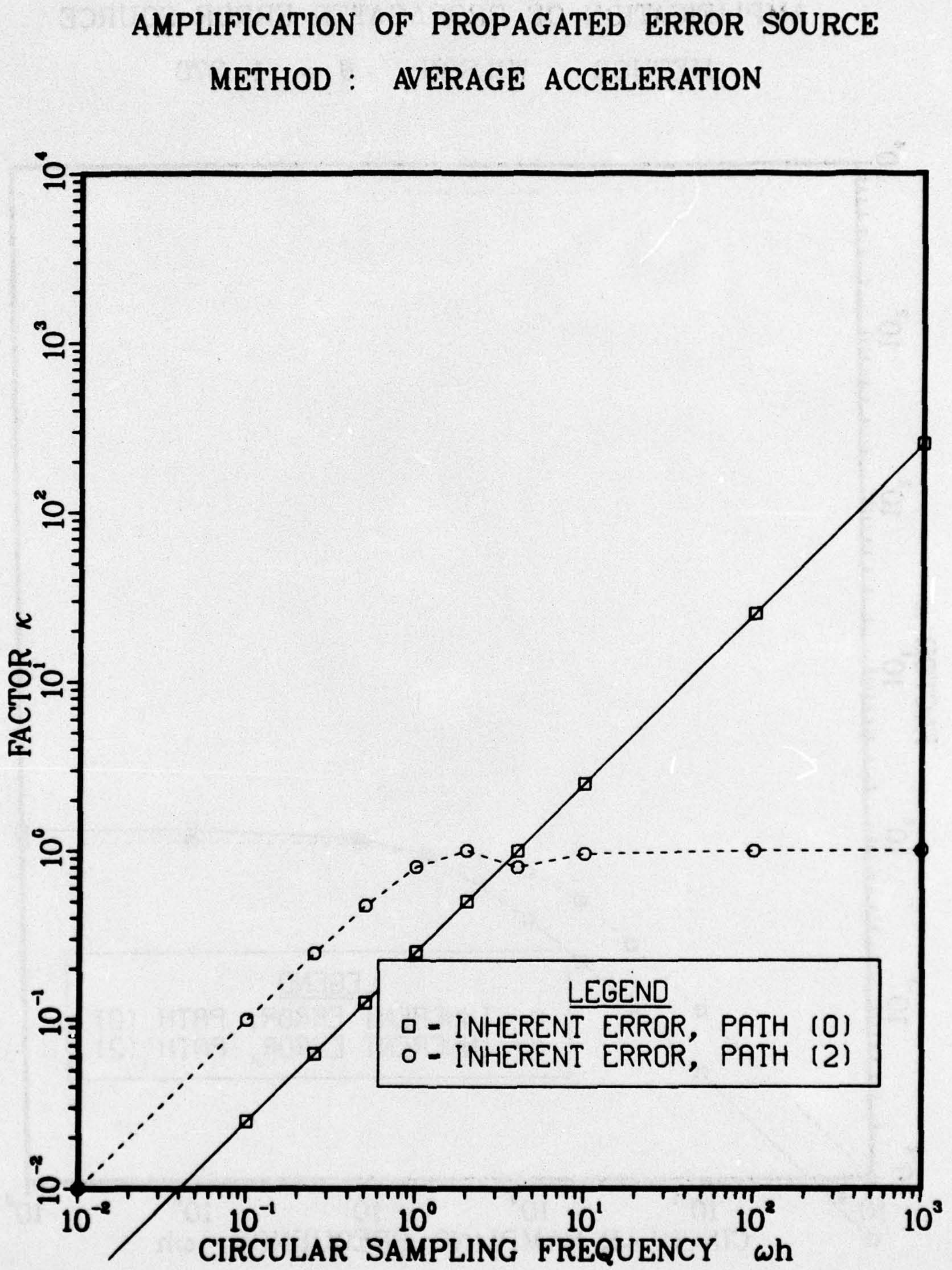


Fig. 23 Amplification spectra of propagated inherent error for the average acceleration method

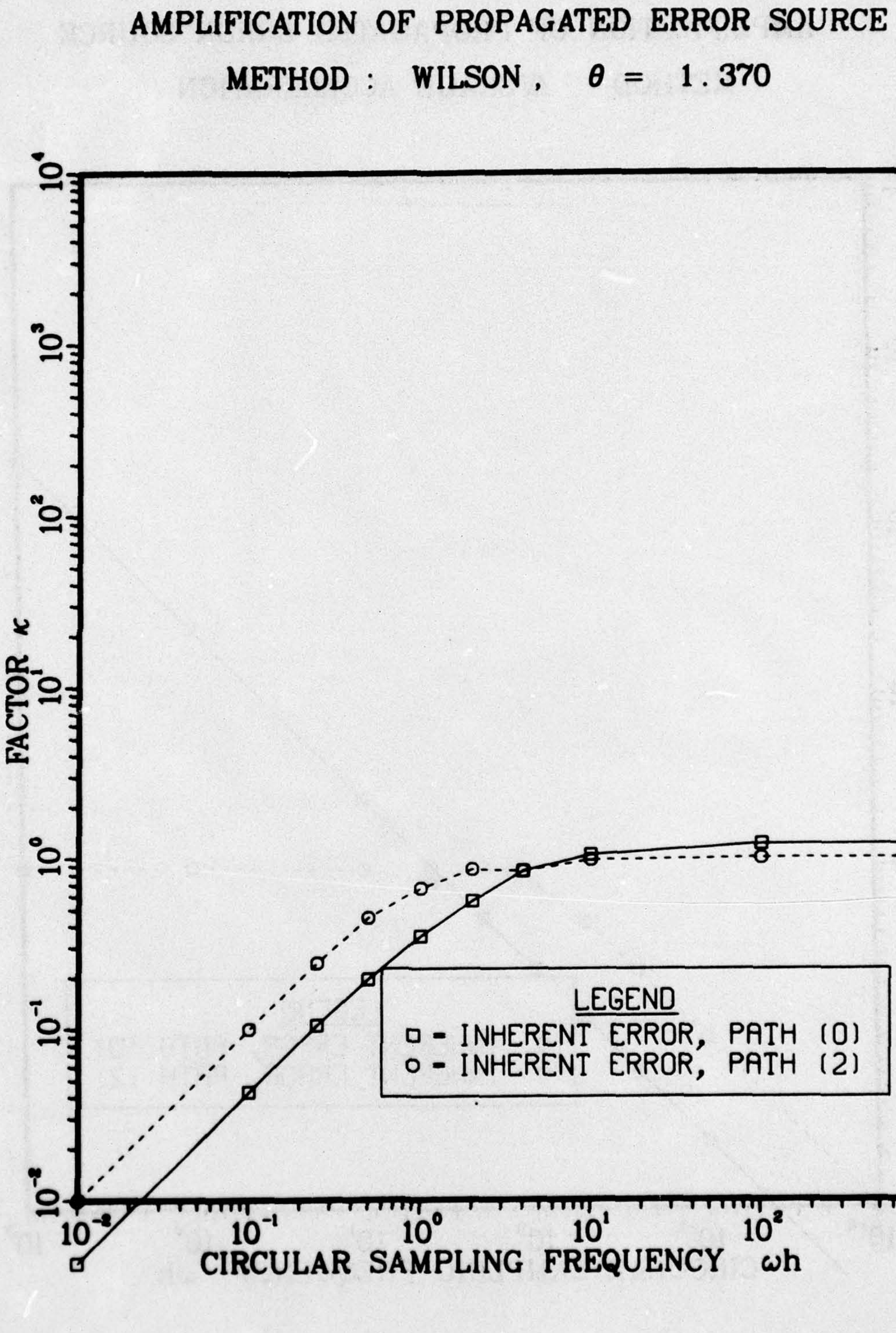


Fig. 24 Amplification spectra of propagated inherent error for Wilson's method with θ = 1.37

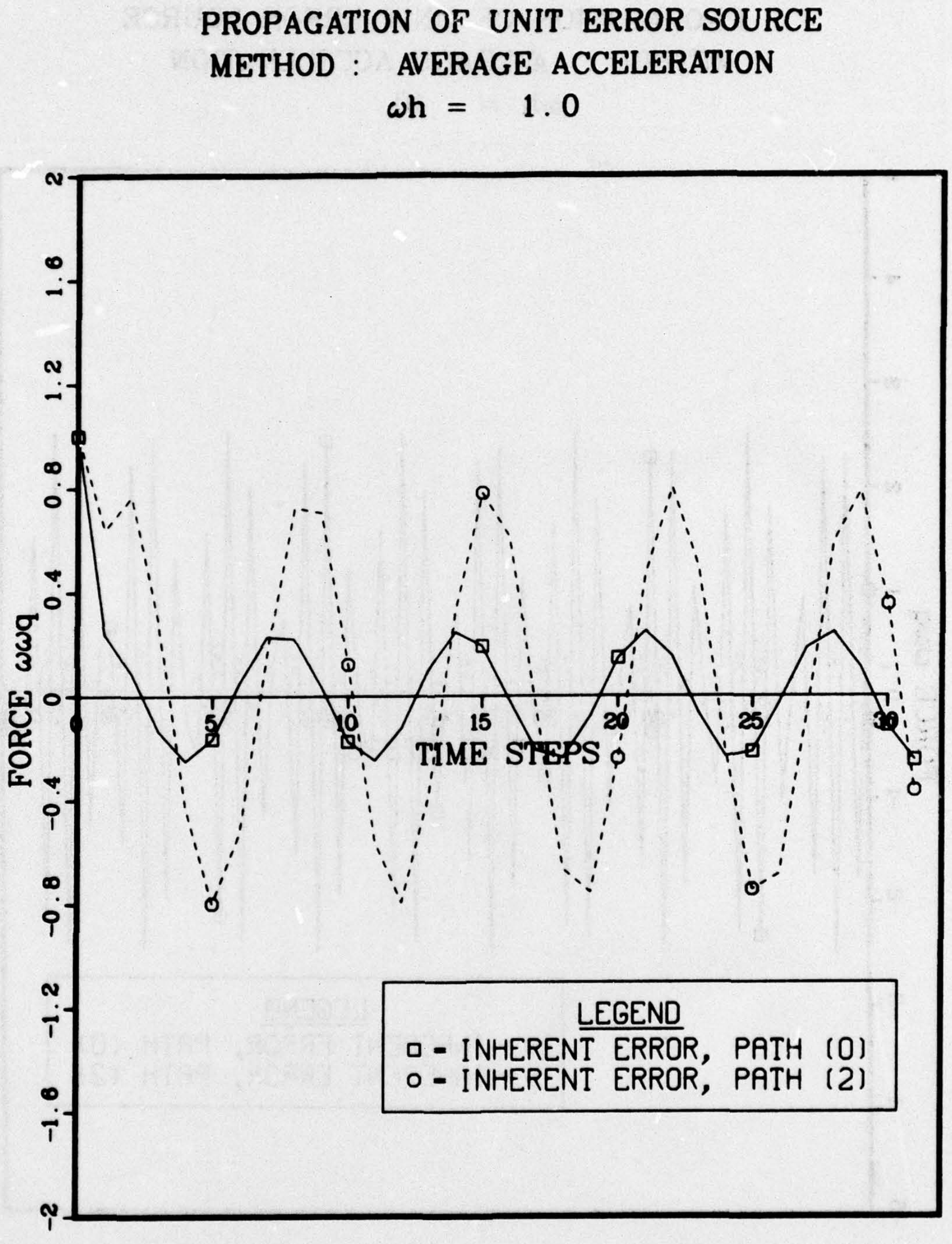


Fig. 25 Propagation history of an intermediate stepsize unit inherent error source for the average acceleration method

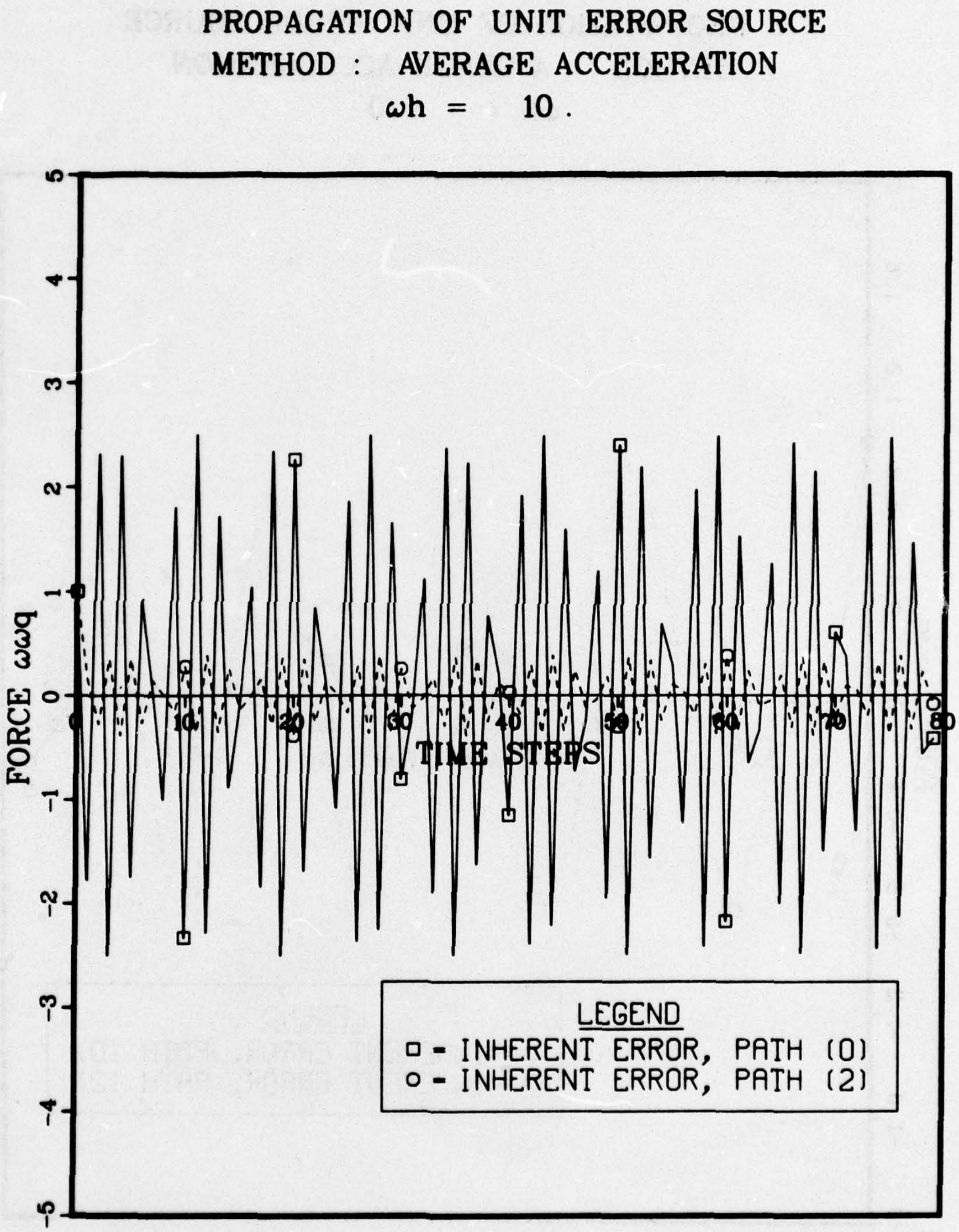


Fig. 26 Propagation history of a large stepsize unit inherent error source for the average acceleration method, showing incipient appearance of numerical beats on path (0)

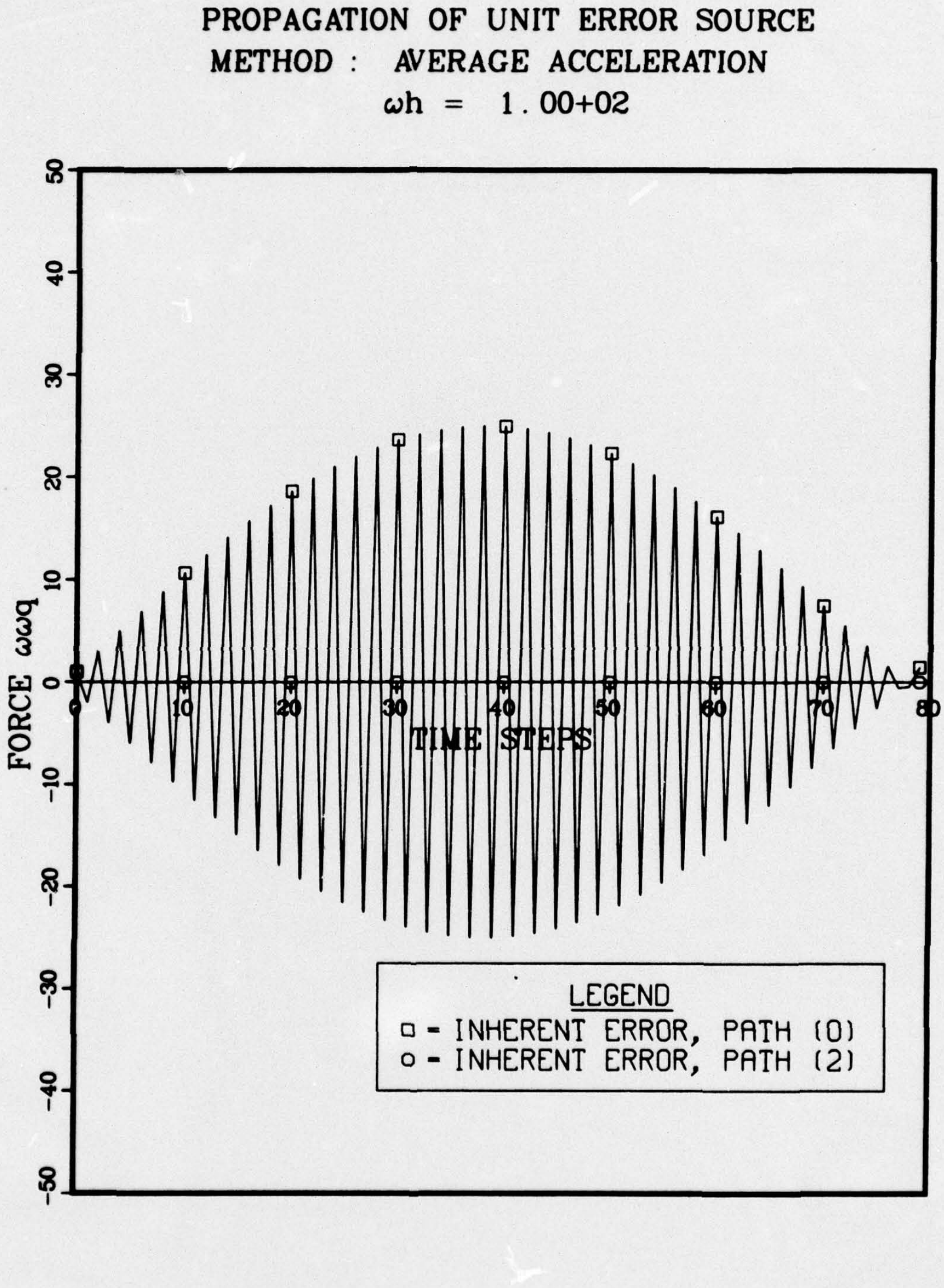


Fig. 27 Propagation history of a large stepsize unit inherent error source for the average acceleration method, showing well-developed numerical beats on path (0)